Non-isometric 3D Shape Registration

Tao Jiang

Thesis

Aug, 2019

Copyright Statement

Abstract

3D shape registration is an important task in computer graphics and computer vision. It has been widely used in the area of film industry, 3D animation, video games and AR/VR assets creation. Manually creating the 3D model of a character from scratch is tedious and time-consuming, and it can only be completed by professional trained artists. With the development of 3D geometry acquisition technology, it becomes easier and cheaper to capture high-resolution and highly detailed 3D geometries. However, the scanned data are often incomplete or noisy and therefore cannot be employed directly. To deal with the above two problems, one typical and efficient solution is to deform an existing high-quality model (template) to fit the scanned data (target). Shape registration as an essential technique to do so has been arousing intensive attention.

In last decades, various shape registration approaches have been proposed for accurate template fitting. However, there are still some remaining challenges. It is well known that the template can be largely different with the target in respect of size and pose. With the large (usually non-isometric) deformation between them, the shear distortion can easily occur, which may lead to poor results, such as degenerated triangles, fold-overs. Before deforming the template towards the target, reliable correspondences between them should be found first. Incorrect correspondences give the wrong deformation guidance, which can also easily produce fold-overs. As mentioned before, the target always comes with noise. This is the part we want to filter out and try not to fit the template on it. Hence, non-isometric shape registration robust to noise is highly desirable in the scene of geometry modelling from the scanned

data.

In this PhD research, we address existing challenges in shape registration, including how to prevent the deformation distortion, how to reduce the foldover occurrence and how to deal with the noise in the target. Novel methods including consistent as-similar-as-possible surface deformation and robust Huber-L1 surface registration are proposed, which are validated through experimental comparison with state-of-the-arts.

The deformation technique plays an important role in shape registration. In this research, a consistent as-similar-as-possible (CASAP) surface deformation approach is proposed. Starting from investigating the continuous deformation energy, we analyse the existing term to make the discrete energy converge to the continuous one, whose property we called as energy consistency. Based on the deformation method, a novel CASAP non-isometric surface registration method is proposed. The proposed registration method well preserves the angles of triangles in the template surface so that least distortion is introduced during the surface deformation and thus reduce the risk of fold-over and self-intersection. To reduce the noise influence, a Huber-L1 based non-isometric surface registration is proposed, where a Huber-L1 regularized model constrained on the transformation variation and position difference. The proposed method is robust to noise and produces piecewise smooth results while still preserving fine details on the target.

We evaluate and validate our methods through extensive experiments, whose results have demonstrated that the proposed methods in this thesis are more accurate and robust to noise in comparison of the state-of-the-arts and enable us to produce high quality models with little efforts.

Acknowledgements

First of all, I would like to express my deepest gratitude to my supervisors Prof. Xiaosong Yang and Prof. Jian J. Zhang for their continued guidance and advices. Their support and help, from research to life, make me feel warm on my academic research path. I am really grateful to meet such supervisors as them, who are always patient to discuss new ideas with me. This work is the result from our countless meetings, especially with Prof. Yang, who encourages me throughout this four years.

Part of the inspirations leading to this work also arise from discussions with Prof. Feng Tian. His critical comments, meticulous attitude and kindly patience are always respectful.

Memorable friendships have been established during my 4 years study in NCCA. Thanks to Prof. Jian Chang, Prof. Lihua You, Dr. Zhidong Xiao and Dr. Hongchun Yu for their help and support. Thanks to Kun Qian, Zhao Wang, Shuang Liu, Li Wang, Yanran Li, Nan Xiang and other friends for the amazing time. Thanks to Jan Lewis, Sunny Choi, Cansu Kurt Green and all gradute school staffs, who handled all the administrative tasks and supported all my visiting.

Gratitude also goes to the Sino UK project, BU PhD Fundings, Santander PGR mobility Award, AniNex project for PhD studentship and financial support of my activities.

Finally, I am greatly indebted to my parents.

Declaration

This thesis has been created by myself and has not been submitted in any previous application for any degree. The work in this thesis has been undertaken by myself except where otherwise stated.

Contents

List of Figures

List of Tables

List of Publications

1. **Jiang, T.**, Yang, X., Zhang, J., Tian, F., Liu, S., Xiang, N. and Qian, K., 2019. Huber-L_1-based non-isometric surface registration. The Visual Computer, 35(6-8), pp.935-948.

2. Xiang, N., Wang, L., **Jiang, T.**, Li, Y., Yang, X. and Zhang, J., 2019, July. Single-image Mesh Reconstruction and Pose Estimation via Generative Normal Map. In Proceedings of the 32nd International Conference on Computer Animation and Social Agents (pp. 79-84). ACM.

3. Barbieri, S., **Jiang, T.**, Cawthorne, B., Xiao, Z. and Yang, X., 2018, August. 3D content creation exploiting 2D character animation. In ACM SIGGRAPH 2018 Posters (p. 50). ACM.

4. **Jiang, T.**, Qian, K., Liu, S., Wang, J., Yang, X. and Zhang, J., 2017. Consistent as-similar-as-possible non-isometric surface registration. The Visual Computer, 33(6-8), pp.891-901.

5. Wang, Z., Liu, S., Qian, R., **Jiang, T.**, Yang, X. and Zhang, J.J., 2016, November. Human motion data refinement unitizing structural sparsity and spatial-temporal information. In 2016 IEEE 13th International Conference on Signal Processing (ICSP) (pp. 975-982). IEEE.

6. Qian, K., **Jiang, T.**, Wang, M., Yang, X. and Zhang, J., 2016. Energized soft tissue dissection in surgery simulation. Computer Animation and Virtual Worlds, 27(3-4), pp.280-289.

Chapter 1

Introduction

1.1 Background

Shape registration is a fundamental problem in computer graphics and computer vision, in which the aim is to find a transformation that best aligns two input shapes. It is an essential algorithm to a wide range of applications, such as, 3D shape reconstruction, statistical shape analysis and shape retrieval. For example, the traditional way of character modelling is inefficient as it usually requires plenty of artist efforts to create a high-fidelity and usable 3D model. Even though the appearance of depth sensors make capturing objects much easier, these data are usually noisy, incomplete and not ready-to-use. To fill this gap, shape registration is introduced to transform an existing high quality model so that it well aligns with the scanned shape.

To guide the transformation in the shape registration, we need to find a deformation mapping that transforms the template shape to the target. The typical classes of deformation mapping and their corresponding geometrical properties are listed in Table 1.1. According to the type of deformation mapping [Floater & Hormann 2005], shape registration is generally categorized into two groups: rigid registration and non-rigid registration. The rigid registration aims to find a rigid-body transformation between two shapes, and thus, it cannot handle deformable (non-rigid) shapes. Non-rigid registration is then categorized into iso-

metric and non-isometric. Isometric registration aims at finding a set of local rigid transformations but lacks local scalability due to its length-preserving property. Non-isometric registration can be further categrated into: equiareal, smooth and similar. Specifically, equiareal registration has scale-preserving property, so is unable to address size difference between the template and the target. In contrast, smooth registration based on smoothness regularization is able to handle size difference. However, it allows piecewise stretching transformation, which can result in shear distortion. Similar registration fits the deformation gradient into a similarity matrix, which is an isotropic scale factor s times a rotation matrix $\mathbf{R}$, which is not only able to handle size difference but also prevents local stretch and shear distortion. Thus, it has been widely used in works [Yamazaki et al. 2013; Yoshiyasu et al. 2014; Papazov & Burschka 2011] to align shapes with different sizes and detail. However, the energies they adopt to constrain the local deformation similarity are not consistent, which tends to produce fold-overs and self-intersection during transformation. Here, consistent indicates that the discrete energy should converge to the continuous case as the discretization refined.

Deformation mapping			df	Property
Rigid			$\mathbf{I}$	shape-preserving
Non-rigid	Isometric		$\mathbf{R}$	length-preserving
	Non-isometric	Equiareal	$\det(df)=1$	scale-preserving
		Smooth	$\min \|df\|^2$	smooth deformation
		Similar	$s\mathbf{R}$	angle-preserving

Table 1.1: *Typical classes of deformation mapping. df is the deformation gradient, s is a scalar,* $\mathbf{I}$ *is an identity matrix,* $\mathbf{R}$ *is a rotation matrix.*

The process of shape registration can be approximately summarized in three steps: firstly, roughly aligning the global positions, scales, orientations between shapes; secondly, searching the correspondences between shapes; finally, register the template to the target according to the correspondences. Finding the correspondences is an essential step for shape registration. According to the survey [Tam et al. 2013], there are some constraints, including features, saliency, envelope of motion, search constraints can be used to assist in finding correspondences for non-rigid registration. The template and the target can be largely different in

size. Since isometric registration is unable to deal with shapes with large size difference, we mainly focus our research on the non-isometric case. In the scope of non-isometric registration, not all the constraints listed above can be applied. Some constraints are suitable for the isometric case but not for the non-isometric. For example, feature and signature constraints can be well defined under isometric circumstance. There are some isometry-invariant features, such as Heat Kernel Signature (HKS) [Sun et al. 2009] and Wave Kernel Signature (WKS)[Aubry et al. 2011], which can be employed to seek for the correspondences as they are invariant under isometric deformation. However, due to the large variations in the pose, size and local details of the shapes for non-isometric case, it is difficult to define such a signature that are invariant to non-isometric transformation. it makes the correspondence searching become a challenging problem for non-isometric registration.

The scanned data usually contain noise, how to exclude the noise and recover the original shape of the target is one goal of shape registration. For the accuracy and robustness of registration, the transformation variation and position difference constraints are usually formulated as a regularization term and a data term, respectively, to measure the smoothness of the neighbouring transformation and the closeness of registration shapes, respectively. Most works [Amberg et al. 2007; Li et al. 2008; Jiang et al. 2017] use the classic squared L_2-norm on both terms (L_2-L_2). However, the regulation term in L_2-norm tends to penalize large transformation variation. It is not suitable for articulated models where large deformation variations exist at their joints. To solve this problem, Yang et al. [2015] propose a sparse non-rigid registration method with L_1 norm applied on the regularization term (L_1-L_2) to allow large transformation discontinuity. The data term in L_2-norm strives to distribute errors evenly, thus fitting the result even on the noisy parts. To tackle this issue, Li et al. [2018] propose a dual sparsity registration approach with L_1-norm applied on both regularization term and data term (L_1-L_1). However, their methods are only limited to isometric registration and the L_1-L_1 model tends to produce piecewise constant artefacts as shown in [Werlberger et al. 2009]. Whereas in the real-life application,

the template and the target can be largely different in shape and size, which means their deformation mapping is non-isometric. Therefore, to devise robust constraints against noise for non-isometric shape registration is highly expected.

1.2 Main challenges

The main challenges for non-isometric 3D shape registration can be summarized as follows:

- **Template quality preservation** Shapes in non-isometric registration can be largely different in respect of size, pose and details. During the registration, the template may undergo large deformation, which makes it susceptible to shear distortion, self-intersection and fold-overs. Obtaining a high-quality, clean and usable shape for application is very challenging.
- **Semantic correspondence** Semantic correspondence is crucial for 3D shape registration. For example, during face registration, features around eyes, mouths and noses should match each other. It is very challenging to find semantic correspondences in the case of non-isometric registration, since these features are prone to be variant under non-isometric deformation.
- **Registration robustness** The scanned target often contains noise and is usually incomplete. It is very challenge to obtain a high-quality result under these circumstances. The noise can easily affect the correspondence searching and the incomplete parts give no correspondence for the template. Without enough correct correspondences, it is difficult to register the template to the target with the shape quality preserved.
- **Less user effort** The registration technique should not rely on a great amount of user input to specify many feature points manually. Otherwise, it would be inefficient and prone to errors for users to operate.

1.3 Research aims

The aim of this research is to tackle these key technique challenges in non-isometric 3D shape registration. The major tasks range from deformation regularization, correspondence searching, shape de-noising and completion, 3D shape registration. This research will propose non-isometric registration methods for fitting a high-quality template shape to a noisy or incomplete target with only little user effort. Techniques devised in this thesis could be applied in 3D character modelling, 3D facial registration and human pose estimation, which would benefit for applications such as film industry, digital games, 3D computer animation and VR/AR interaction.

1.4 Research Objectives

In order to achieve the aim, following objectives need to be accomplished:

- **Literature Review**: review and investigate current research on 3D geometric deformation, correspondence searching, de-noising in image and shape processing, 3D shape registration. Identify the limitation of current approaches in non-isometric situations.
- **Non-isometric 3D Geometric Deformation**: design a novel non-isometric 3D shape deformation technique that is able to handle large deformation with shape quality well preserved.
- **Correspondence Searching**: design an effective method to find reliable correspondences between non-isometric shapes which should be robust enough against noisy and incomplete target.
- **Robustness**: design robust constraints to constrain the shape deformation so that the template quality will be well maintained even given a noisy target.
- **Little User Effort**: with little user effort only, our methods could accomplish the non-isometric registration automatically.
- **Non-isometric 3D Shape Registration**: equipped with the

above approaches, design a robust non-isometric registration method with little user effort, which is robust against shear distortion, large deformation and noise so that the template quality is well-preserved.

1.5 Contributions

There are several major contributions towards different tasks in this PhD research:

- We propose a novel shape deformation method, called consistent as-similar-as-possible (CASAP) deformation. It fits local transformation into scale and rotation, which is not only able to handle large deformation, but also reduces the shear distortion. Moreover, contrast to as-similar-as-possible (ASAP) deformation method, the energy in our deformation technique is consistent, which means the discrete energy can be converged to the continuous case as the discritization refined.

- With CASAP energy as deformation regularization, we further propose a non-isometric surface registration approach. It not only produces more accurate fitting results with little user input, but also preserves angles of triangle meshes and allows local scales to change. Furthermore, a coarse-to-fine strategy is proposed to further improve the robustness and efficiency of our approach.

- Taking local geometrical feature descriptors into account, we propose a new matching energy to find more reasonable correspondent pairs between template and target models.

- We propose a Huber-L_1 based non-isometric registration method regularized on transformation variation and position difference. The Huber-L_1 model is solved by the alternating direction method of multipliers (ADMM) with each energy term being represented in matrix form. The proposed model is robust to noise and produces piecewise smooth results with the target's fine details being well preserved.

1.6 Structure of the following chapters

The following part of this thesis contains 7 more chapters:

- **Chapter 2** Literature review on the related research topics, including 3D geometric deformation, correspondence searching, shape denoising and 3D shape registration.
- **Chapter 3** Presents a novel deformation method, which is not only capable of addressing shapes with large difference in size but also ascertain a consistent energy to reduce the chance of fold-over and self-intersection.
- **Chapter 4** Presents a novel non-isometric surface registration method, which is not only able to handle large deformation but also reduces the occurrence of fold-over and shear distortion during transformation.
- **Chapter 5** Presents a Huber-L_1 based non-isometric surface registration, which is robust to noise and produces piecewise smooth results.
- **Chapter 6** Conclusion and future plan.

Chapter 2

Literature Review

The registration process usually consists of three steps: aligning the template with the target globally; finding correspondence between shapes; deforming the template towards the target via deformation technique. This chapter will review the key technologies for shape registration including 3D geometric deformation, 3D shape correspondence and 3D shape registration. In addition, as the target usually comes along with noise, we will also review the robust registration methods aiming to tackle the noise issue.

2.1 3D geometric deformation

Geometric deformation is a fundamental and deeply researched topic in the field of computational geometry. It has important applications in a lot of areas, for example computer animation, film industry, video game and manufacturing. A multitude of works have been published aiming at this issue. According to the survey [Botsch & Sorkine 2008], based on whether the deformation problem can be formulated as a linear system, the deformation methods can be generally divided into two categories:

- Linear deformation;
- Nonlinear deformation.

2.1.1 Basic concept

Before diving into the review, we first introduce some basic concepts and notations of deformation referred to in this research. As we all know, the objects existing in nature is in the form of continuity. However, in computer they have to be represented discretely as computers can only address discrete signals. In computer, the shapes (both surfaces and volumes) are usually discretized into discrete units. We name these discrete units as discrete cells. In general, the surface can be discretized into triangles or quadrilaterals, while the basic 3D discrete cells for volume are the tetrahedron, quadrilateral pyramid, triangular prism and hexahedron. In this research, we mainly focus on triangulation surfaces. Although shapes represented in computer are in discrete forms, the deformation energy should converge to the continuous case as the discritization is refined, which we name as consistent property. By consistent, we mean for different discrtizations of the same shape, their energies should be consistent when the same deformation is applied. This is particulary useful in the case of a poor triangulation that has large variation in the size of the triangles.

For triangulation mesh, the global deformation can be decomposed into each triangle or each vertex. Moreover, these triangles or these vertices are not independent from each other, they are connected by neighboring edge sets. There are three kinds of discrete cell for triangulation mesh: triangle, spokes, spokes-and-rims (Figure 2.1).

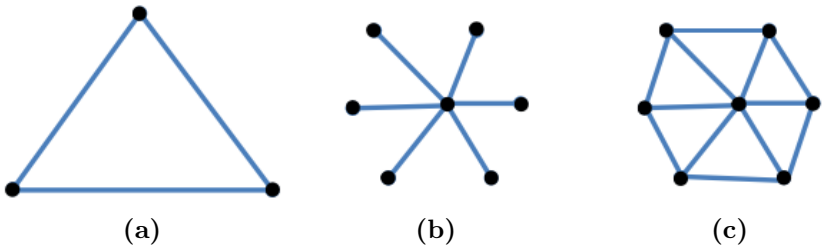


Figure 2.1: *Different cells for triangulation mesh: (a) triangle; (b) spokes; (c) spokes-and-rims;*

2.1.2 Linear deformation

Linear means we use a linear system of equations to solve a global quadratic variational minimization problem. The advantage of linear deformation methods are three-folds: efficient, solving a linear system of equation is quite efficient, especially when the associated linear system is sparse; robust, the solved quadratic energy has a unique global minimum; smooth, the global energy minimization guarantees smooth and $\mathcal{C}^1$ continuous surface deformations compared the non-linear deformation. According to the target the deformation applied onto, the linear deformation methods can be categorized into linear surface based deformation and linear space deformation.

Linear surface-based deformation

The linear surface-based deformation roughly falls into three categories: shell-based deformation, multi-scale deformation and differential coordinates based deformation.

Shell-based deformation method minimizes the elastic energy subject to user-defined boundary constraints. The elastic energy measures how much the object has been deformed from its initial configuration, which is the main requirement for physically based surface deformations. For two-manifold surfaces, the elastic energy considers local stretching and bending within the object. Terzopoulos et al. [1987]; Celniker & Gossard [1991] propose physically-based deformation methods minimizing stretching and bending under deformation constrains, which corresponds to thin-shell models of non-planar rest states. Deformation based on a discretization of variational bending energy minimization is mathematically understood and yields smooth and tangent-continuous deformations [Guskov et al. 1999; Kobbelt et al. 1998; Botsch & Kobbelt 2004; Bickel et al. 2008]. However, for these approaches all computations and linearizations are performed with respect to a fixed reference mesh, large deformations might lead to shape distortions and detail loss.

To preserve surface details, the above methods require a multi-scale decomposition, which splits a surface into a smooth base surface (low fre-

quencies) and displacement vectors (high frequencies). Multiresolution deformation changes the smooth base surface and adding the details back onto it then yields the desired multi-scale deformation [Kobbelt et al. 1999]. In particular, Kobbelt et al. [1998] introduce a mesh deformation technique by solving a constrained minimization of the thin-plate energy at a desirable coarse resolution. The user specifies deformation constraints through a handle polygon. Original mesh details are added back to the resulting smooth mesh to produce a final solution. This technique only gives the user limited control over the mesh shape through sparse constraints on the handle polygon. The rest of the mesh geometry is uniquely determined by the minimization. Displacement volumes [Botsch & Kobbelt 2003] encode the high frequencies by prism elements enclosed between the original and the base surface, which avoids detail distortion, but comes at the considerably higher cost of a non-linear detail reconstruction. Although both representations (displacement vectors/volumes) can be combined with any underlying deformation technique, the required multiscale decomposition can become quite difficult for geometrically or topologically complex models.

To avoid the multi-scale decomposition, other methods modify differential surface properties instead of its spatial coordinates, and then solve a linear Poisson system for a deformed surface with the desired differential coordinates [Lipman et al. 2004; Sorkine et al. 2004; Yu et al. 2004; Zayer et al. 2005; Lipman et al. 2005]. The methods of Yu et al. [2004]; Zayer et al. [2005] first manipulate the original surface gradients and then deformed the surface to match the target gradient field in the least-squares sense. As a consequence, these methods work well for rotations, but are insensitive to translations: Adding a translation to a given deformation does not change its gradient, and thus has no influence on the resulting surface gradients. But since even pure translations induce local rotations of tangent planes, these methods are counter-intuitive for modifications containing large translations. In contrast, the shape editing approach of Sorkine et al. [2004] aims to preserve the differential coordinates or Laplacian coordinates. It implicitly solves for local rotations of vertex neighborhoods, but due to linearizations their method has

problems with large rotations, as shown in their follow-up paper [Lipman et al. 2005]. In that paper, Lipman et al. minimize bending by preserving relative per-vertex orientations. They solve a linear system for per-vertex orientations, and reconstruct vertex positions. Since the first system does not consider position constraints, their technique also neglects the connection between translations and rotations, it exhibits the same translation-insensitivity as gradient-based methods.

Linear space deformation

The need for low-dimensional control of deformation fields was identified early in computer graphics. Among the first approaches was Free-Form Deformation [Sederberg & Parry 1986], which relied on regular lattices to specify spatial deformations. It parameterizes a space deformation with a 3D lattice and provides an efficient way to apply coarse deformations to complex shapes. However, achieving a fine-scale deformation may require a detailed, hand-designed control lattice [Coquillart 1990; MacCracken & Joy 1996] and a large amount of user manipulation. Although more intuitive control can be provided through direct manipulation [Hsu et al. 1992], the user is still restricted by the the expressibility of the FFD algorithm. With their Wires concept, Singh & Fiume [1998] present a flexible and effective space deformation algorithm motivated by armatures used in traditional sculpting. A collection of space curves tracks deformable features of an object, providing a coarse approximation to the shape and a means to deform it. Singh & Kokkevis [2000] generalize this concept to a polygon-based deformer. Botsch & Kobbelt [2005] use triharmonic radial basis functions for real-time freeform shape editing. An incremental least-squares method is introduced to approximately solve the involved linear systems in a robust and efficient manner.

Cage-based deformation methods were an important step forward, because control polytopes offer much better adaptability to the input shapes. The underlying theme of many cage-based methods is to generalize barycentric coordinates from simplices to general polytopes. Mean value coordinates (MVC) for closes polyhedrons [Ju et al. 2005] offer many desirable properties and can be calculated using closed-form ex-

pressions, but are not fully shape-aware. This shortcoming has been addressed by harmonic coordinates [Joshi et al. 2007]. Lipman et al. [2007] introduce positive mean value coordinates (PMVC). Unlike the MVC, the modified coordinates are unconditionally positive, and require only a local computation. The methods mentioned above are affine-invariant and not shape-preserving. Lipman et al. [2008] introduce Green coordinates for closed polyhedral cages. It is not only dependent on vertex-based basis, but also on the cage faces, which leads to space deformations with a shape-preserving property. While many new intriguing coordinates and their underlying mathematical properties have been studied in recent years [Hormann & Sukumar 2008; Weber et al. 2011; Li & Hu 2013], the problem common to all cage-based method remains: the design of control cages requires experience with polygonal modeling: the cage should be close to the shape and have enough density to represent the shape.

2.1.3 Nonlinear deformation

The surface deformation problem is inherently non-linear, it requires deducing local rotations of the surface based on position displacements. Therefore, a linear method can only provide an approximate result, or a compromise must be made in terms of the problem setup, e.g., requiring more complex interactive input from the user, for example adding more user handle constraints or increasing the cage density. The inherent limitations of linear methods motivated us to investigate non-linear deformation techniques. Nonlinear deformation can also be classified into nonlinear surface-based deformation and nonlinear space deformation.

Nonlinear surface-based deformation

Pyramid coordinates [Kraevoy & Sheffer 2004, 2005] can be considered as the nonlinear versions of Laplacian coordinates, leading to differential coordinates invariant under rigid motions, which can be used for deformation as well as for morphing. Huang et al. [2006] employ a nonlinear version of the volumetric graph Laplacian, which also features nonlinear volume preservation constraints. In order to increase the performance and efficiency of the optimization, they use a subspace approach: the

original mesh is embedded in a coarse control cage, and the optimization is performed on the cage vertices while considering the constraints from the original mesh vertices in a least-squares manner. An alternative approach to subspace methods is the handle-aware isoline technique of [Au et al. 2007]. In a preprocessing step one constructs a set of iso-lines of the geodesic distance from either the fixed regions or the handle regions. For each of these iso-lines, a local transformation for a Laplacian-based deformation is found by a nonlinear optimization. The number of required iso-lines is relatively small, which guarantees an efficient numerical optimization and thereby allows for interactive editing. Shi et al. [2007] combine Laplacian-based deformation with skeleton-based inverse kinematics, which allows for easy and intuitive character posing, featuring control of lengths, rigidity and joint limits, but it in turn requires a complex cascading optimization for the involved nonlinear energy minimization. Botsch et al. [2006] propose a nonlinear version of the shell-based minimization of bending and stretching energies. The surface is modeled as a thin layer of triangular prisms, which are coupled by a nonlinear elastic energy. During deformation the prisms are kept rigid, which allows for a robust geometric optimization. A hierarchical optimization is used to increases the computational efficiency.

Sorkine & Alexa [2007] propose a surface deformation based on the as-rigid-as-possible (ARAP) energy. The ARAP energy measures the local deviation of the differential of a mapping between two shapes from rigidity. The neighboring edge sets adopted in this work are spokes, which requires a positive weighting scheme to guarantee the correct minimization of the energy. Chao et al. [2010] take into account all the opposite edges in the triangles incident to a vertex, the neighboring edge sets in their work are spokes and rims, which guarantee correct minimization of the energy even if the weights are negative. However, the discretization of [Chao et al. 2010] is only consistent for volumetric case with tetrahedron cells in 3D or parameterization with triangle edge sets in 2D, it is not consistent for the surface case using spokes and rims edge sets in 3D. In order to come up with a consistent discretization for surface in 3D, Levi & Gotsman [2015] introduce a new ARAP-type energy, named

SR-ARAP (ARAP with smooth rotations), they add a bending term in the ARAP energy to enable the discretization consistent, which achieves results with quality that competes with the volume deformation.

Nonlinear space deformation

Sumner et al. [2007] compute detail-preserving space deformations by formulating an energy functional that explicitly penalizes deviation from local rigidity by optimizing the local deformation gradients to be rotations. In addition to static geometries, their method can also be applied to hand-crafted animations and precomputed simulations. Botsch et al. [2007] extend the PriMo framework [Botsch et al. 2006] to deformations of solid objects. The input model is voxelized in an adaptive manner, and the resulting hexahedral cells are kept rigid under deformations to ensure numerical robustness. The deformation is governed by a nonlinear elastic energy coupling neighboring rigid cells. Another class of approaches uses divergence-free vector fields (the divergence of the vector fields is zero) to deform shapes [Angelidis et al. 2006; von Funck et al. 2006]. The advantage of those techniques is that by construction they yield volume-preserving and intersection-free deformations. As a drawback, it is harder to construct vector fields that exactly satisfy user-defined deformation constraints.

Müller et al. [2005] present meshless deformations based on shape matching. Instead of using the vertex connection information, they divide the set of vertices into clusters. Afterwards, affine transformations can be applied on each cluster. Based on this work, Rivers & James [2007] introduce fast lattice shape matching for real-time deformation. Since the original method can be very slow for stiff models - per-vertex costs scale cubically with region width - they exploit the inherent summation redundancy of shape matching and provide large-region matching at constant per-vertex cost. With this approach, large lattices can be simulated in linear time complexity.

Jacobson [2013] introduces the harmonic, biharmonic, triharmoic equations w.r.t. surface displacement fields correspond to minimizers of the

Dirichlet, Laplacian, Laplacian gradient energies. They offer a detail derivation to obtain a linear system to solve this second-order elliptic partial differential equation. Weber et al. [2007] use harmonic functions that are provably monotonic and bounded but have only C^0 smoothness near constrained boundary. Bounded biharmonic weights are proposed in Jacobson et al. [2011] to minimize the Laplacian energy subject to bound constraints. They spread the influences of the controls in a shape-aware and localized manner, even for objects with complex and concave boundaries. Jacobson et al. [2010] use mixed finite elements to provide a discretization for biharmonic and triharmonic equations on meshes. They factorize the original equations into low-order equations by introducing additional variables, and introduce lumping mass matrices to eliminate unneeded variables. Their minimizers exhibit increasing orders of continuity but lose the maximum principle and show oscillations. Jacobson et al. [2012] propose smooth shape-aware functions with controlled extrema. They provide a framework for minimizing quadratic energies on manifolds while constraining the solution to obey the maximum principle in the solved region.

Other methods

In this subsection, we review the data-driven and learning based deformation methods. Gao et al. [2016] propose a new rotation-invariant deformation representation and a novel reconstruction algorithm to accurately reconstruct the positions and local rotations simultaneously. Based on this, they further propose a data-driven mesh deformation approach by exploiting knowledge in the example models. Gao et al. [2019] propose a sparse data-driven deformation method that automatically selects a smaller number of deformation modes to compactly describe the desired deformation and solve the overfitting problem. Kurenkov et al. [2018] introduce free-form deformation as a differentiable layer to enable 3D data manipulation in their network. however, this network outputs a set of points rather than a deformed mesh. Wang et al. [2019] propose an end-to-end network architecture for mesh deformation. Given a source mesh and a target model, their model can generate plausible deformed

meshes by inferring pervertex displacements while keeping the source mesh connectivity fixed.

2.1.4 3D geometric deformation summary

3D geometric deformation is a crucial part of shape registration as it serves as a regularization to attract the template onto the target. This section investigates the state of the art linear and nonlinear deformation techniques, and compares them in terms of handling large deformation and consistency. The linear methods can only approximate the local rotations, which are commonly seen in surface deformation. Nonlinear approaches are able to accurately determine local rotations. However, some of them do not allow local scales, which are required in addressing large local deformation. Although some methods allow local scale but they are not consistent, which is prone to fold-over and self-intersection, especially in the case of poor triangulation. In this research, a novel deformation algorithm will be proposed to address large difference in size, as well as to guarantee the energy consistent to minimize the occurrence of fold-over and self-intersect.

2.2 3D shape correspondence

Establishing a meaningful shape correspondence is a fundamental task in applications such as object recognition, statistical shape modeling, shape morphing and deformation transfer. In this section, we will give an overview of the correspondence research from three point of views: similarity-based correspondence, rigid alignment and non-rigid alignment.

2.2.1 Similarity-based correspondence

One of the most fundamental ways of computing correspondence is to estimate the similarity between pairs of shape elements or feature points collected from the shapes and derive a correspondence from those estimates, which is sometimes called the feature matching approach. The

elements are commonly characterized by shape descriptors. There are plenty of shape descriptors that can be computed for each shape primitive and used in conjunction with correspondence algorithms for 2D surfaces or 3D point sets (Table 2.1). Note that the surface descriptor listed here are not scale-invariant, which means they may change with the variation of shape's scale.

Descriptor	Type of dataset
Shape context [Belongie et al. 2001]	Point sets
Spin images [Johnson 1997]	Oriented points
Multi-scale features [Pauly et al. 2003]	Oriented points
Curvature maps [Gatzke et al. 2005]	Surfaces
Integral invariants [Manay et al. 2006]	Surfaces
Spherical harmonics and wavelets [Laga et al. 2006]	Surfaces
Salient geometric features [Gal & Cohen-Or 2006]	Surfaces
Part-aware metric [Liu et al. 2009]	Surfaces
Heat Kernel Signature [Bronstein & Kokkinos 2010]	Surfaces

Table 2.1: *Shape descriptors can be used for shape correspondence.*

A correspondence is obtained by selecting assignments between pairs of elements while optimizing an objective function composed of two terms. The first term seeks to maximize the similarity between the descriptors of corresponding elements, while the second term seeks to minimize the distortion that would be introduced in the shapes if they were deformed to align their corresponding elements. However, the second term can be estimated without explicitly aligning the shapes. Ideally, satisfying these objectives should translate into a solution that is geometrically or semantically meaningful. Such a solution is typically obtained with a standard optimization method (e.g., quadratic programming).

Feature matching can be applied in any context where it is possible to compute a set of descriptors for the elements. Example applications include registration of 3D scans [Castellani et al. 2008] and deforming surfaces [Anguelov et al. 2005b], or skeleton matching [Biasotti et al. 2006]. Moreover, this approach is not restricted to its own domain and can be combined with alignment-based approaches to provide a proper initialization to these methods [Rusinkiewicz & Levoy 2001], or to restrict the size of the solution space [Gelfand et al. 2005; Kin-Chung Au et al.

2010; Chang & Zwicker 2008; Aiger et al. 2008].

2.2.2 Rigid alignment

Rigid alignment is to find a global geometric transformation that aligns the shapes. One example application is the rigid alignment of geometry scans used for shape acquisition. The goal here is to capture a real-world static 3D shape and obtain its digital representation. However, it may not be possible to capture the entire object in a single scanning pass due to self-occlusions and physical constraints of the scanner, so it might become necessary to acquire multiple scans and optimally align them to reconstruct the full object [Turk & Levoy 1994; Rusinkiewicz & Levoy 2001; Gelfand et al. 2005; Aiger et al. 2008]. The key characteristic of the rigid alignment problem is that the objects do not change from one scanning pass to another. Thus, it is assumed that each scan can be transformed with a single rigid transformation in order to align it perfectly with the other scans. Rigid transformations comprise translations and rotations, and one of their important characteristics is that they reside in a low-dimensional space.

Scan alignment is just one example of many applications that rely on the assumption of rigidity in the datasets. If the input shapes are given as sets of 3D points, the problem of rigid alignment can be posed as: for each point set, find the rigid transformation that maximizes the number of points in the set that align to points in the other sets. This goal is usually dependent on a threshold that indicates when two points are close enough and can be considered as matching to each other [Irani & Raghavan 1999]. Since finding the best aligning transformation might be a complex task, the feature matching approach can be brought in to aid in the search for the optimal alignment.

2.2.3 Non-rigid alignment

Sometimes it might be necessary to lift the assumption that each scan can be perfectly aligned with a rigid transformation, e.g., when large amounts of noise are present in the scans. More significant examples

of datasets that cannot be perfectly aligned with a rigid transformation include the correspondence of articulated shapes [Elad & Kimmel 2003; Anguelov et al. 2005b; Jain et al. 2007; Chang & Zwicker 2008; Huang et al. 2008], where certain parts of the shapes can be bended independently, the correspondence of anatomical shapes (e.g, organs) [Audette et al. 2000], which can deform in an elastic manner and introduce stretching to localized portions of the shape, and finally the correspondence between shapes with different geometries but that represent a same class of objects or which have parts that are semantically related [Allen et al. 2003; Zhang et al. 2008]. In the latter case, the alignment problem can be concluded as establishing a correspondence between shapes that can differ in both local stretching and bending.

To this end, it becomes necessary to add more degrees of freedom (DOF) to how the shapes can be brought into correspondence. This can be achieved by generalizing two aspects of the problem. First, non-rigid (possibly non-linear) transformations can be taken into consideration, e.g., thin-plate splines [Chui & Rangarajan 2003]. Secondly, these transformations can be applied separately to local portions of the shape. For example, the transformation applied to a shape can be represented as a set of displacement vectors (one per shape vertex) [Pauly et al. 2005]. Then, finding the best transformation amounts to computing the displacements that bring each vertex in correspondence with the target shape. The distinction to the rigid case is that the DOF of geometric transformations being considered is now inherently high-dimensional.

Tevs et al. [2009] present a global deformable matching approach based on a novel RANSAC-like randomized sampling algorithm. They draw the potential correspondences from a probability density function. Isometric consistency is formulated as a likelihood to adjust the posterior probability such that more reliable correspondences are determined in the next iteration. Tevs et al. [2011] propose a shape matching algorithm based on the novel concept of entropy-based planned random sampling. The algorithm automatically adapts to the input characteristics and chooses an optimized sampling strategy for any given object. Ovsjanikov et al. [2012] propose a novel representation maps between

isometric shapes to find correspondences between generic functions defined on the shapes. Pokrass et al. [2013] extended this approach to pose the problem of finding intrinsic correspondence between near-isometric deformable shapes as a problem of sparse modeling. Kovnatsky et al. [2013] proposed constructing common approximate eigenbases for multiple shapes using approximate joint diagonalization algorithms. In such bases the correspondence is represented by an approximately diagonal matrix, which allows to solve the linear system of equations for diagonal elements only. Kovnatsky et al. [2015] model the correspondence between the spaces of functions on two manifolds as a matrix and pose the functional correspondence problem as matrix completion with manifold geometric structure. These methods only work well among quasi-isometric shapes. Huang et al. [2017] present a view-based convolutional network that produces local, point-based shape descriptors. Their network is trained such that geometrically and semantically similar points across different 3D shapes are embedded close to each other in descriptor space. Wang et al. [2020] propose a novel framework for computing descriptors for characterizing points on three-dimensional surfaces. In their work, two types of shape descriptors are computed. First, the non-learned descriptor WEDS is computed using graph wavelets to decompose the Dirichlet energy on a surface. Second, WEDS can be refined by multiscale GCN to yield a learned descriptor. Wang et al. [2018] present a novel deep learning framework that derives discriminative local descriptors for 3D surface shapes. These learning based methods are powerful, however, a lot of manual annotation is required and it only works well on the similar models which the network has trained on.

2.2.4 3D shape correspondence summary

In the process of shape registration, finding correspondence is the priori step before deforming the template, it determines the goal positions of the deformed template. This section reviews the methods solving rigid and non-rigid correspondence problems. Rigid alignment is only effective with static shapes. For deformable shapes, similarity-based correspondence methods work well between isometric shapes, as the descriptors it

adopted can be isometrically invariant such as Heat Kernel Signature. However, there is no such shape descriptor which is invariant to large non-isometric deformation. Non-rigid alignment assumes the template can be transformed onto the target in a specified type of deformation, however, the space of solution to large deformation itself is quite high dimensional and hard to find. In this research, we design a novel matching energy to choose correspondence between non-isometric shapes, the correspondences are first chose based on local geometric feature descriptors and then extended to local optimal candidates.

2.3 3D shape registration

3D registration is an active research topic in computer graphics and computer vision. Given a template and a target, the goal of 3D registration is to find a mapping between them in order to optimally transform the template onto the target. Shape registration methods can fall roughly into 3 classes: rigid, isometric, non-isometric.

2.3.1 Rigid registration

The purpose of rigid registration is to find a global rigid transformation that aligns two shapes. It is quite similar to rigid alignment in last subsection, just has a different final goal. The most classic algorithms for rigid registration are Iterative Close Point (ICP) Besl & McKay [1992] and its variants Rusinkiewicz & Levoy [2001]. ICP alternates between searching correspondences by choosing closest points on the target w.r.t. the points on the template and computing a rigid transformation to response the correspondences. The results closely depend on the initial position and orientation of the shapes, large distance between them tends to produce unstable correspondence and ICP is susceptible to converge to local minimum. Aiming to solve this problem, Gelfand et al. [2005] initially approximate the optimal transformation via feature matches, this approximation can be fed into ICP to improve the convergence. Another shortage of ICP is that it is sensitive to noise and outliers which often appear in 3D scans. To tackle this issue, Bouaziz et al. [2013] put

forward a new variant of ICP algorithm with sparsity-inducing norms, successfully eliminating the affect of the noise and outlier and obtaining superior results.

2.3.2 Isometric registration

Isometric registration is to solve a local rigid transformation for each vertex on the template in order to align the template onto the target. Li et al. [2008] propose an isometric registration algorithm for partial scans of deforming shapes within a single non-linear optimization. The rigid energy term they adopted penalizes the deviation of each local transformation from a pure rigid motion, which is only composed of rigid and orientation transformation. Huang et al. [2008] regard isometric registration as an optimization problem. To solve this problem, they alternate between correspondence searching and deformation calculation. Under the assumption of local rigid deformation, robust correspondences are obtained via a pruning mechanism based upon consistency in geodesic distance. Chang & Zwicker [2008] convert the isometric registration problem into a discrete labeling problem. They first sample the motion explicitly, which provides a priori set of possible rigid transformation between articulate parts of shapes, and then find an optimal assignment of transformation to each part of shapes. Tevs et al. [2009] propose a global isometric matching approach based on a novel RANSAC-like randomized sampling algorithm. This method is robust to topological noise and is able to output matching alternatives by sampling the space of plausible solutions. Maron et al. [2016] approximate the global minimum of the procrustes matching problem with convex relaxation for non-rigid isometric and near-isometric shape matching problem. The advantage of isometric registration is that they can be achieved automatically, but they are incapable of handling models with different sizes.

2.3.3 Non-isometric registration

Non-isometric registration is a wide range of topics including all kinds of registration that are not isometric. Here we only focus on techniques

related to our work: similar registration, affine registration and smooth registration.

Similar registration

Similar registration is to preserve the local similarity of shapes, it remains the angle of intersection of every pair of the intersecting arcs unchanged during deformation process. Sorkine et al. [2004] offer a linear approximation of similarity matrix to make deformed Laplacian coordinates consistent. However, this method only works well under small rotations as the approximation removes the quadratic term. Thus it cannot handle large rotation. Yamazaki et al. [2013] extend ARAP energy to *as-similar-as-possible* (ASAP) energy with spokes edge sets. The work in [Papazov & Burschka 2011] is a variation of shape matching [Müller et al. 2005] called similarity shape matching. Although these techniques utilize similar mapping to enable them to address size difference and shear distortion, they do not consider the smoothness regularization, which results in they are incapable of handling large changes in pose or shape. Yoshiyasu et al. [2014] incorporate smooth regularization into the total energy, however, it is an unweighted energy, which does not take into account the impact of the length of edges. Moreover, the edge sets they adopt are spokes, which leads to an inconsistent energy.

Affine registration

Affine registration allows an affine transformation for each template vertex, which will have a larger solution space due to more freedom the affine matrix has. Amberg et al. [2007] employ a locally affine regularisation to each vertex and smooth the difference in the transformation of neighbouring vertices. The method is robust to incomplete data and is able to handle a wide range of initial conditions. Yang et al. [2015] propose a sparse non-rigid registration method with an l_1-norm regularized model for transformation estimation, they allow an affine transformation for each point in the template to cover a wide range of non-rigid deformations. Affine registration is able to handle shapes in different size, but the affine matrix with more DOF may also introduce shear and

distortion, which easily lead to fold-over and intersection.

Smooth registration

Smooth registration is based on harmonic mapping and its varieties or other smoothness regularization. By smooth we mean the adjacent element should have similar deformation behavior. Weise et al. [2009] provide a complete integrated system for live facial puppetry which not only is able to track high-resolution facial expression in real time but also can transfer the expression to another face. They minimize a membrane energy on the displacement vectors to smooth the deformation on the template. Yeh et al. [2011] present a novel approach for template-based 3D model fitting , they approximate the input geometry with a linearized biharmonic surface, and then use iterative Laplacian editing and a local surface flattening mechanism to avoid foldovers. There are some works [Allen et al. 2003; Amberg et al. 2007] based on other smoothness regularization whose purpose are to make the deformation between neighbours as smooth as possible, this idea is similar to the bending term added in Levi & Gotsman [2015]. Similar to affine registration, smooth registration still has the same advantage and problem. In addition, it requires many landmarks from user input to obtain a good initial shape.

2.3.4 3D shape registration Summary

In summary, 3D shape registration with large size and detail difference is a challenging problem mainly because of the requirement of large deformation which is susceptive to fold-over and intersection. This section investigates all kinds of registration methods and their capabilities. Rigid registration is only suitable for static models. Isometric registration can work with deformable objects but only limit to isometric shapes or near isometric shapes. Affine and smooth registration are able to handle models with different sizes, but they are too weak against shear distortion. Although similar registration can overcome these disadvantages, the energy itself is not consistent, which easily leads to fold-over and self-intersection when the triangulation mesh is poor. In this research, a novel registration technique will be present to solve large shape

variation as well as minimize the chance of fold-over occurrence.

2.4 Robust registration

The registration robustness can be achieved by applying different norms on constraints [Yang et al. 2015; Li et al. 2018]. Before introducing the robust registration methods, we first explain the related norms and their properties.

2.4.1 Norm definition

In this research, we mainly focus on three different norms: L_2-norm, L_1-norm and Huber-norm. The graphs of function of single variable in different norms is depicted in Figure 2.2. The L_2-norm is calculated as the square root of the sum of the squared variable values. Generally, a squared L_2-norm is applied on constraint energy to simplify the calculation. Without special instruction, we refer the L_2-norm as the squared L_2-norm in this research. The L_1-norm is calculated as the sum of the absolute values of each variable. Huber-norm is defined between L_2-norm and L_1-norm. It behaves like an L_2-norm below a certain threshold ϵ and like an L_1-norm above. To explain the properties of these norms, without loss of generality, we restrict the optimization problem in 2D domain. The goal of optimization is to find the optimal solution under constraints of two variables in a specific norm. The curve of 2D constrains in different norms can be found in Figure 2.3. For L_2-norm, the optimal solution to $\sqrt{(x^2 + y^2)} = c, c >= 0$ can be evenly location on the circle curve. The cost energy is evenly distributed in every dimension. The L_2-norm has a property of smoothness as it tries to minimize the energy in every dimension equally. For L_1-norm, the optimal solution to $|x| + |y| = c, c >= 0$ can only be location at the corner points of the square. The L_1-norm has a property of sparsity as the cost energy is mainly concentrated in few specific dimension. As its definition, the Huber-norm has L_2-norm like property below threshold ϵ and has L_1-norm like property above.

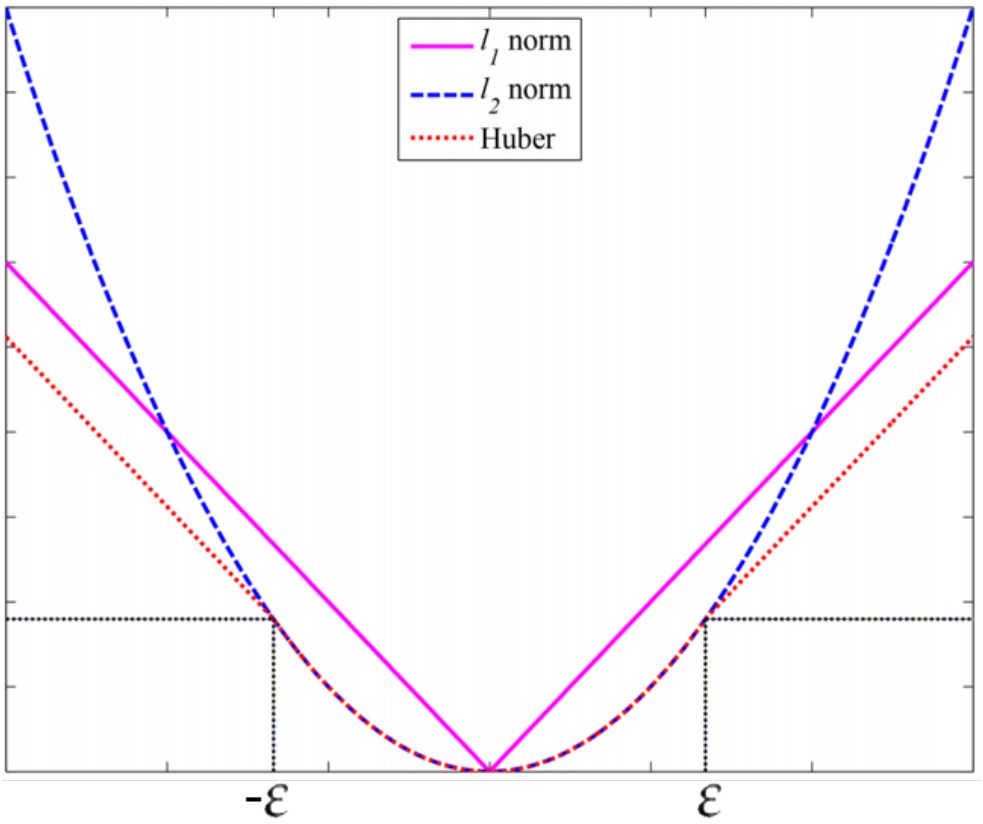


Figure 2.2: *The graphs of function of single variable in different norms. The horizontal axis represents independent variable while the vertical axis represents dependent variable.*

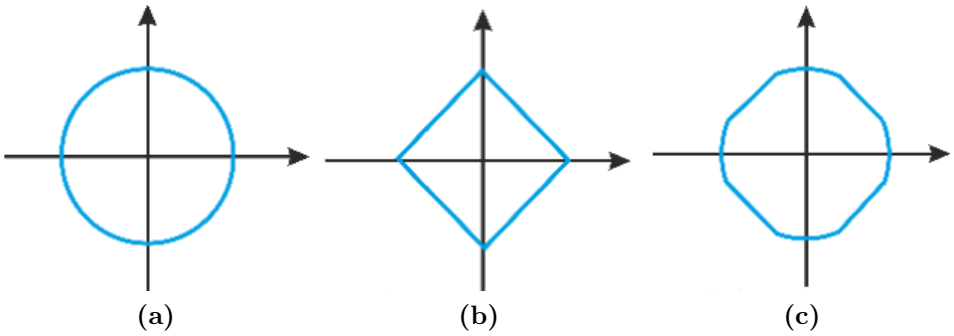
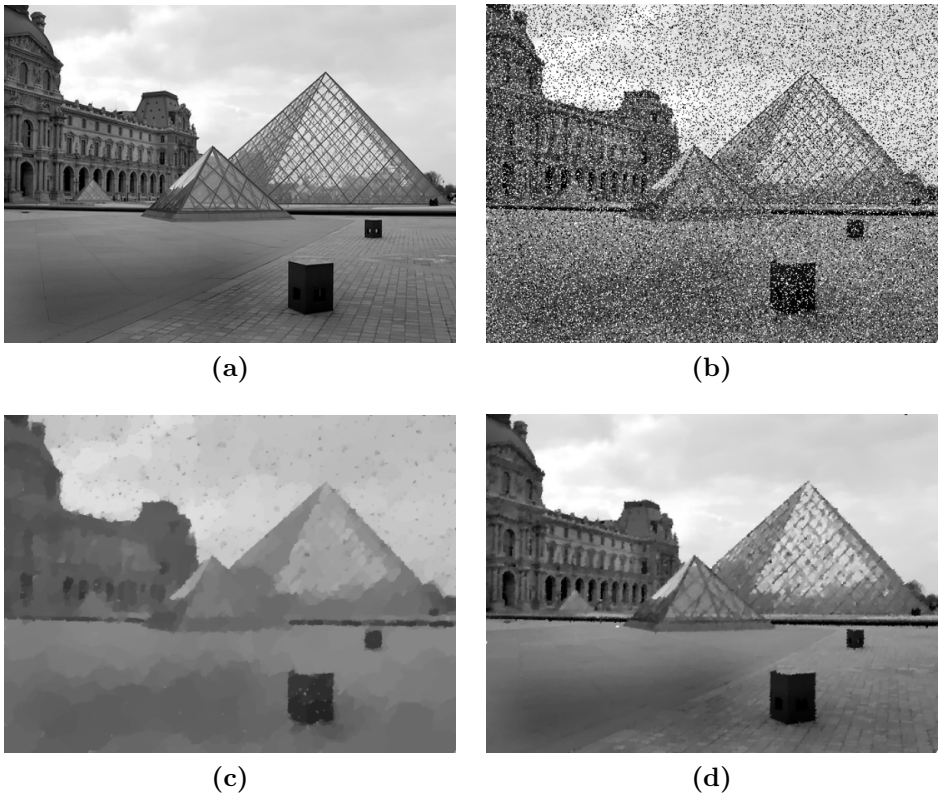


Figure 2.3: *2D constraints in different norms: (a) L_2-norm; (b) L_1-norm; (c) Huber-norm;*

2.4.2 robust registration methods

For the accuracy and robustness of registration, the transformation variation and position difference constraints are usually formulated as a smoothness term and a data term respectively to measure the smoothness of the neighbouring transformation and the closeness of registration shapes, respectively. Most works [Amberg et al. 2007; Li et al. 2008; Jiang et al. 2017] use the classic squared L_2-norm on both constraints (L_2-L_2). However, the smoothness term in L_2-norm tends to penalize large transformation variation. It is not suitable for articulated models where large deformation variations exist at their joints. This could also be seen in image processing where discontinuities are allowed to highlight the sharp edges in image denoising [Chambolle & Pock 2011]. The classic model in image denoising is ROF model Rudin et al. [1992], where the total variation term is an L_1-norm based regularization and the data term is in squared L_2-norm (L_1-L_2). Based on ROF model, Yang et al. [2015] propose a sparse non-rigid registration method with an L_1-norm regularized on the smoothness term. However, ROF tends to produce over-regularized results as the L_2-norm strives to distribute errors evenly, thus fitting the result evenly on the noisy parts. To tackle this issue, L_1-L_1 model [Zach et al. 2007] is proposed to efficiently remove the outliers while preserving fine details (Figure 2.4). Based on L_1-L_1 model, Li et al. [2018] propose a dual sparsities registration approach on both position and transformation sparsity, allowing the positional error to concentrate on small regions. However, as pointed out in [Werlberger

et al. 2009], L_1-L_1 model tends to produce piecewise constant results as shown in Figure 2.5.



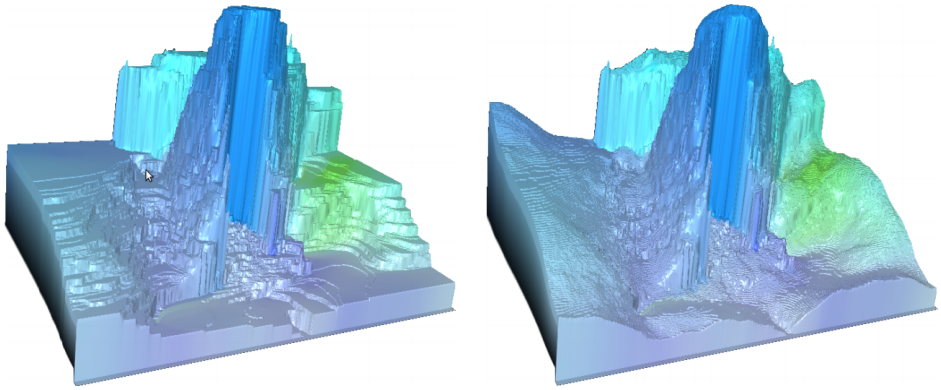
(a) **(b)**

(c) **(d)**

Figure 2.4: *Image denoising in the case of impulse noise. (a) shows the* 500×375 *input image and (b) is a noisy version which has been corrupted by* 25% *salt and pepper noise. (c) is result of the ROF model. (d) is the result of the* L_1-L_1 *model. Note that the* L_1-L_1 *model is able to remove the noise while still preserving some small details.*

2.4.3 Robust registration summary

In the summary, 3D shape registration which is robust against noise is highly desirable because the noise is commonly seen in the target. This section investigates all kinds of norms and their properties. L_2-norm based methods strive to minimize the energy in every dimension. They are easily affected by outliers. Taking advantage of the sparsity property of L_1-norm, L_1-L_1 model based methods are proposed to exclude the outlier influence while still preserving the fine details. However, these methods tend to produce piecewise constant artifacts. In this research, we will propose a novel model, which is not only robust to noise but also



(a) **(b)**

Figure 2.5: *Comparing (a) the staircasing afflicted L_1-L_1 model and (b) the Huber-L_1 model on the Dimetrodon dataset.*

produce piecewise smooth results while still preserving fine details.

Chapter 3

Consistent As-Similar-As-Possible Surface Deformation

3D geometric deformation plays a key role in computer graphics and computer vision, which has been widely used in industries like computer animation, video games and film production. Large deformation is a challenging problem as it tends to produce shear distortion, which may lead to poor quality results during deformation, such as fold-overs and self-intersections. Moreover, for non-isometric deformation, geometric scalable is an essential property as the size of the shape may change in non-isometric deformation. This will make the problem even more complicate since it involves more unknown variable to solve.

In the last two decades, there have been plenty of methods proposed on shape deformation aiming to tackle these challenges. However, these approaches either are unable to handle large deformation or suffer from fold-overs or self-intersections. In this chapter, we will propose a novel deformation algorithm, called consistent as-similar-as-possible surface deformation (CASAP), which is not only capable of addressing shapes with large difference in size but also ascertain a consistent energy to reduce the chance of fold-over and self-intersection. The experiments have shown that our new surface deformation scheme competes with the qual-

ity of the volumetric ASAP deformation, which is regarded as a ground truth of the consistent ASAP deformation method.

3.1 Notations

Before stepping into the problem, we first introduce some basic concepts and notations of deformation which will be used in this research. We denote by $\mathcal{S}$ a triangle mesh. The piecewise linear geometric embedding of $\mathcal{S}$ is defined by the vertex positions $\mathbf{p} \in \mathbf{R}^3$. Assume $\mathcal{S}$ is being deformed into $\mathcal{S}'$ that has the same connectivity and a different geometric embedding $\mathbf{p}'$. The discrete cell corresponding to vertex i is denoted by $\mathcal{C}_i$ and its deformed version $\mathcal{C}'_i$. $\mathcal{E}_i$ is the corresponding edge sets in the cell $\mathcal{C}_i$.

3.2 Related methods

In this section, we will introduce three related deformation methods: as-rigid-as-possible (ARAP), smooth rotation enhanced as-rigid-as-possible (SR-ASAP), as-similar-as-possible (ASAP). We will compare our method with these three methods in the experiments.

3.2.1 ARAP

Sorkine & Alexa [2007] propose an as-rigid-as-possible energy to measure the local rigidity variation, in which the neighboring edge sets adopted are spokes. Given two meshes $\mathcal{S}$ and $\mathcal{S}'$ consisting of vertices $\mathbf{p}$ and $\mathbf{p}'$ respectively, the discrete ARAP energy between these two meshes is defined as:

$$E_{ARAP}(\mathcal{S}, \mathcal{S}') = \sum_i \sum_{j \in \mathcal{N}_i} w_{ij} \|(\mathbf{p}'_j - \mathbf{p}'_i) - \mathbf{R}_i(\mathbf{p}_j - \mathbf{p}_i)\|^2, \qquad (3.1)$$

where $\mathcal{N}_i$ is the one-ring neighbour of vertex i, w_{ij} are weighting coefficients, $\mathbf{R}_i \in RO(3)$ are optimal local rotation matrices. In the shape deformation setup, deforming a mesh $\mathcal{S}$ involves fixing handle points and solving for the rest of the $\mathbf{p}'$ by minimizing (3.1). The goal of minimiz-

ing the ARAP energy is to find a mesh $\mathcal{S}'$ that is transformed from the source mesh $\mathcal{S}$ in a locally rigid way. More specifically, the differential of deformation mapping for a local cell should be optimally a rotation matrix.

3.2.2 SR-ARAP

Although ARAP shape deformation gained popularity, the energy is not consistent as it lacks a term measuring bending energy. Suppose we have a mesh consisting of two triangles, no matter how much we bend these two triangles, the ARAP energy will not be changed (Figure 3.1).

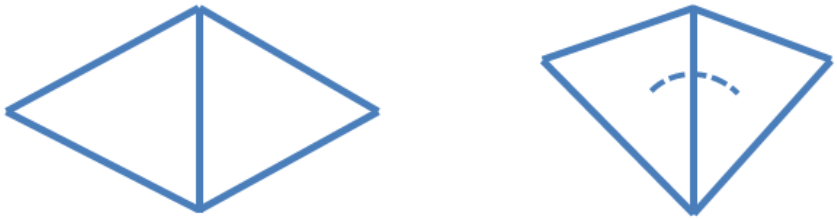


Figure 3.1: *The weakness of ARAP method: the ARAP method will not measure the bending energy. Bending the triangles on the surface will not affect the ARAP energy.*

To overcome this weakness, Levi & Gotsman [2015] introduce a new ARAP-type energy, named SR-ARAP (ARAP with smooth rotation). The new energy used for surface deformation is consistent and can produce results that compete with the volumetric deformation methods. SR-ARAP adds a smoothing term to the ARAP surface energy to penalize the smooth difference. The SR-ARAP energy is defined as:

$$E_{SR-ARAP}(\mathcal{S},\mathcal{S}') = \sum_i \sum_{(j,k)\in\mathcal{E}_i} w_{jk}\|(\mathbf{p}'_j - \mathbf{p}'_k) - \mathbf{R}_i(\mathbf{p}_j - \mathbf{p}_k)\|^2 + \alpha A \sum_{\mathcal{E}_l\in\mathcal{N}(\mathcal{E}_i)} w_{il}\|\mathbf{R}_i - \mathbf{R}_l\|_F^2, \tag{3.2}$$

where $\mathcal{N}(\mathcal{E}_i)$ are the neighboring cells of $\mathcal{E}_i$; α is a weighting coefficient; A is the area of the whole mesh surface, which is used to make the energy scale invariant (scaling the edges by s, would scale the first term by s^2, which is the scale of A in the second term); w_{il} is scalar weight; $\|\cdot\|_F$

denotes the Frobenius norm. The second term is the bending energy, which penalizes the rigidity difference between a cell and its neighboring cells. In this way, they have made up the missing bending energy in ARAP to form a *consistent* one.

3.2.3 ASAP

Although SR-ARAP energy overcomes weaknesses in the ARAP surface deformation, which achieves the consistent discretization for the surfaces, it is still not capable of handling non-isometric deformation since it aims to preserve the local rigidity of each cell. To solve this problem, Yamazaki et al. [2013] propose an as-similar-as-possible deformation method. They add an isometric scale factor s_i to the deformation gradient of each local spokes cell , which allows local scalability on each discrete cell and thus is able to handle deformation with different sizes. The ASAP energy is defined as:

$$E_{ASAP}(\mathcal{S}, \mathcal{S}') = \sum_i \sum_{(j,k)\in\mathcal{E}_i} w_{jk} \|(\mathbf{p}'_j - \mathbf{p}'_k) - s_i \mathbf{R}_i(\mathbf{p}_j - \mathbf{p}_k)\|^2, \tag{3.3}$$

However, similar to ARAP energy, ASAP energy is also not consistent, which results in susceptive to fold-overs and self-intersections (Figure 3.4).

3.3 Our method

In order to tackle the problems mentioned above, we propose a novel surface deformation energy, called consistent as-similar-as-possible (CASAP) energy, by introducing local scale to the deformation gradient in the previous SR-ARAP energy. It has advantages of SR-ARAP and ARAP, which is not only able to handle deformation with different sizes, but also reduce the occurrence of fold-over and self-intersection.

3.3.1 Consistent as-similar-as-possible surface deformation

Assuming we are deforming a mesh $\mathcal{S}$ into $\mathcal{S}'$ in an as similar as possible way, unlike [Sorkine & Alexa 2007; Yamazaki et al. 2013; Yoshiyasu et al. 2014] regarding spokes as the cell, the cell chosen in our method is spokes and rims (denoted as $\mathcal{E}_i$) in order to arrive at an analysable energy. By analyzable, we mean the discrete energy can be analysed and compared with the continuous energy. In the case of spokes and rims, the discrete energy can be calculated as the continuous energy integrated over the entire 1-ring of triangles of a given vertex [Chao et al. 2010]. If the deformation $\mathcal{C}_i \rightarrow \mathcal{C}_i'$ is similar, then there exists a scale factor $s_i > 0$ and a rotation matrix $\mathbf{R}_i$ such that

$$\mathbf{p}_j' - \mathbf{p}_k' = s_i \mathbf{R}_i (\mathbf{p}_j - \mathbf{p}_k), \forall (j,k) \in \mathcal{E}_i, \tag{3.4}$$

where $\mathcal{E}_i$ consists of the set of edges incident to vertex i (the spokes) and the set of edges in the loop (the rims) of vertex i in the surface mesh $\mathcal{S}$. When the deformation is not similar, we can still find the best approximating scale factor s_i and rotation $\mathbf{R}_i$ by minimizing a weighted cost function

$$E(C_i, C_i') = \sum_{(j,k)\in\mathcal{E}_i} w_{jk} \|(\mathbf{p}_j' - \mathbf{p}_k') - s_i \mathbf{R}_i (\mathbf{p}_j - \mathbf{p}_k)\|^2, \tag{3.5}$$

where w_{jk} are edge weighting coefficients. We chose the cotangent weights for w_{jk} in order to achieve a consistent discretization of the membrane like energy [Pinkall & Polthier 1993].

In order to measure the similarity of a deformation of the whole mesh, we sum up over the deviations from similarity per cell which yields following ASAP energy functional:

$$\begin{aligned} E_a(\mathbf{p}') &= \sum_i E(\mathcal{C}_i, \mathcal{C}_i') \\ &= \sum_i \sum_{(j,k)\in\mathcal{E}_i} w_{jk} \|(\mathbf{p}_j' - \mathbf{p}_k') - s_i \mathbf{R}_i (\mathbf{p}_j - \mathbf{p}_k)\|^2. \end{aligned} \tag{3.6}$$

According to [Botsch & Sorkine 2007], the main requirement for physically-based surface deformations is an elastic energy that is able to measure how much the object has been deformed from its initial configuration, which means it can measure both stretching deformation and bending deformation. However, the ASAP energy we obtained so far is not elastic yet. In fact, the ASAP energy only measures the stretching deformation but not the bending deformation. Inspired by [Levi & Gotsman 2015] we incorporate the smooth regularization into (3.6) to make up the bending measurement leading us to an elastic ASAP energy:

$$\begin{aligned} E_d(\mathbf{p}') &= E_a(\mathbf{p}') + E_b(\mathbf{p}') \\ &= \sum_i \Big(\sum_{(j,k)\in\mathcal{E}_i} w_{jk} \|(\mathbf{p}'_j - \mathbf{p}'_k) - s_i \mathbf{R}_i(\mathbf{p}_j - \mathbf{p}_k)\|^2 \\ &+ \alpha A \sum_{\mathcal{E}_l \in \mathcal{N}(\mathcal{E}_i)} w_{il} \|\mathbf{R}_i - \mathbf{R}_l\|_F^2 \Big), \end{aligned} \tag{3.7}$$

where $\mathcal{N}(\mathcal{E}_i)$ are the neighboring cells of $\mathcal{E}_i$; s_i is the scale factor of cell i; α is a weighting coefficient; A is the area of the whole mesh surface, which is used to make the energy scale invariant; w_{il} are scalar weights; $\|\cdot\|_F$ denotes the Frobenius norm. We still choose cotangent weights for w_{il} for constructing consistent bending energy. The second term E_b we add is the bending energy [Levi & Gotsman 2015], which penalizes the rotation difference between a cell and its neighboring cells. In this way, we have made up the missing bending measurement in ASAP energy to form an consistent elastic energy (Figure 3.2).

3.3.2 As-similar-as-possible volumetric deformation

If we integrate the as-similar-as-possible energy over the tetrahedron in a 3D shape, we can easily obtain a consistent as-similar-as-possible discrete energy [Chao et al. 2010]. We propose an as-similar-as-possible volumetric deformation method (VASAP) in order to provide the pseudo ground truth for the SR-ASAP surface deformation method.

The VASAP energy is very similar to the ASAP energy, the only

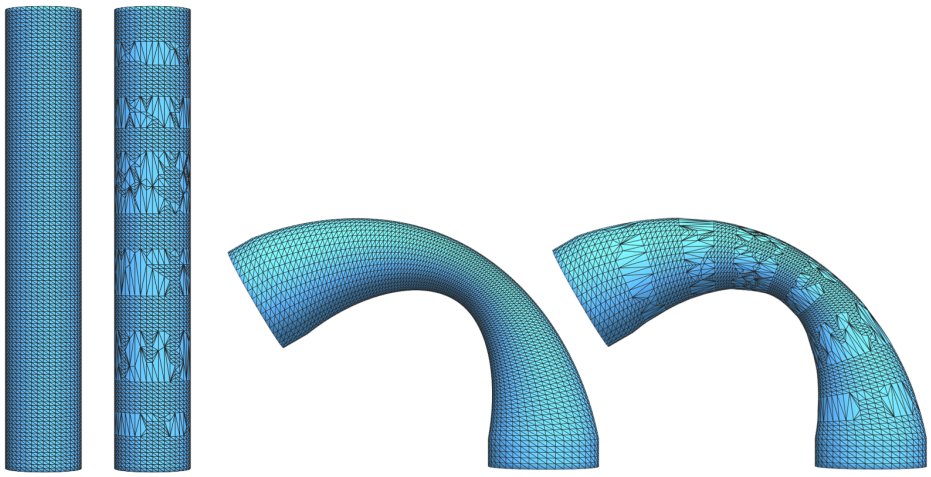


Figure 3.2: *CASAP deformation on the same object with different discretization results in very similar qualitative behaviors.*

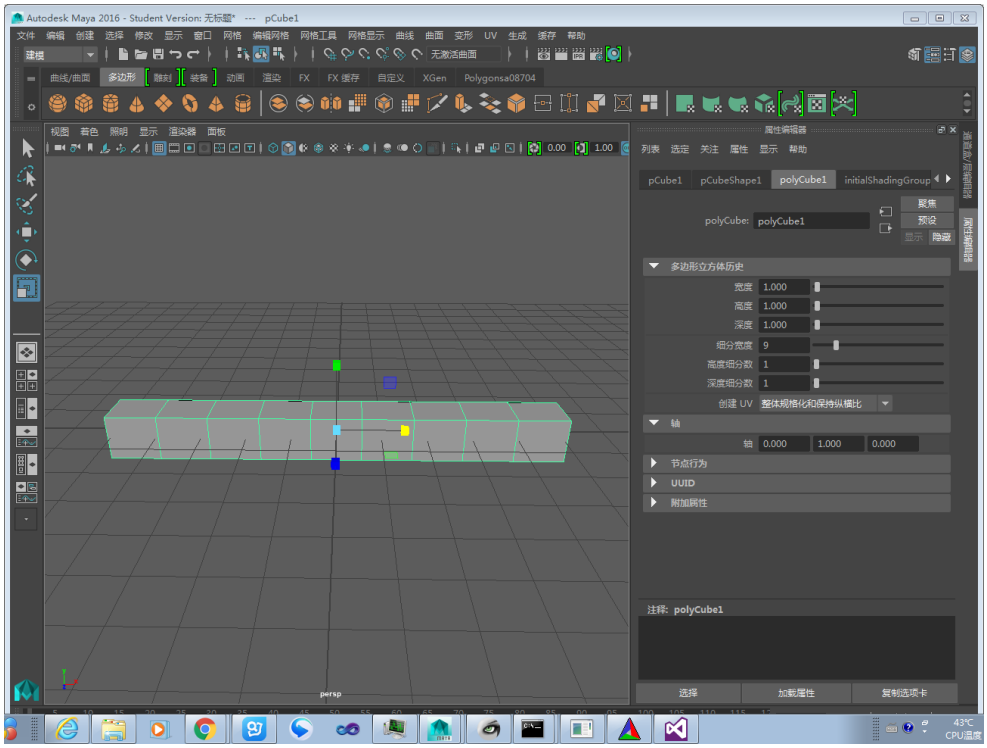
difference is that the local cell $\mathcal{E}_i$ in VASAP is a tetrahedra not spokes:

$$E_{VASAP}(\mathcal{S}, \mathcal{S}') = \sum_i \sum_{(j,k)\in\mathcal{E}_i} w_{jk} \|(\mathbf{p}'_j - \mathbf{p}'_k) - s_i \mathbf{R}_i(\mathbf{p}_j - \mathbf{p}_k)\|^2. \tag{3.8}$$

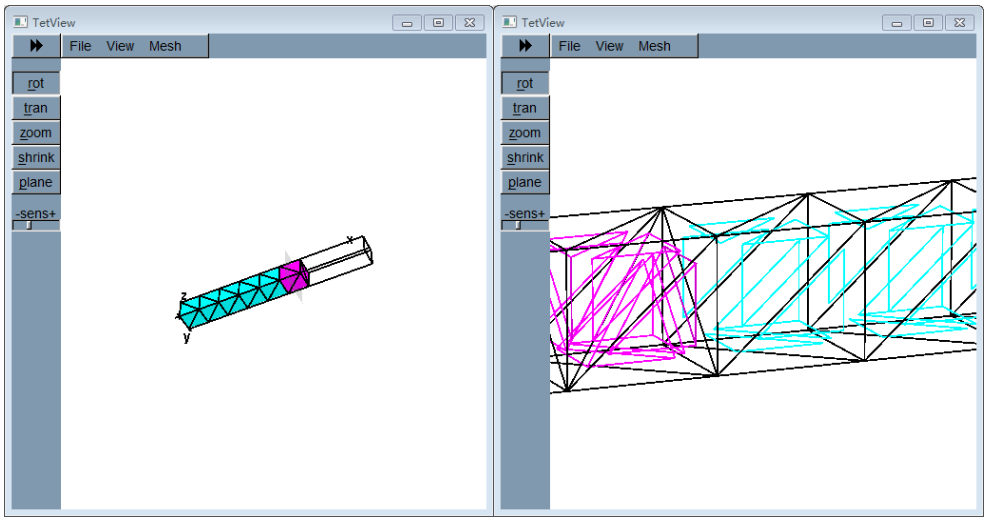
In order to use the VASAP, we need to tetrahedralize the mesh first. We resort to TetGen [Si 2015] to do that. TetGen is a program to generate tetrahedral meshes of any 3D polyhedral domains. TetGen generates exact constrained Delaunay tetrahedralizations, boundary conforming Delaunay meshes, and Voronoi partitions. It has many options, parts of which can be seen in Table 3.1. We use "-pqY" options in our program to preserve the original input surface mesh. We first edit our polygon in the modeling software like MAYA (Figure 3.3a) and then TetGen is utilized to tetrahedralize the polygon. Finally, the generated tetrahedron mesh can be viewed in TetView (Figure 3.3b, 3.3c).

3.4 Optimization

In this section, we introduce the optimization algorithm to minimize the CASAP energy in (3.7). Note that except the vertex positions $\mathbf{p}_i$ are unknown, s_i and $\mathbf{R}_i$ in (3.7) are also unknown for each vertex. We employ



(a)



(b) (c)

Figure 3.3: *The original polygon and its tetrahedralized mesh (a) The original polygon edited in MAYA; (b,c) The generated tetrahedron viewed in TetView.*

the alternating optimization scheme following [Sorkine & Alexa 2007; Yamazaki et al. 2013; Levi & Gotsman 2015] to solve them respectively. Each iteration consists of a local step followed by a global step. In local

Options	Descriptions
-p	Tetrahedralizes a piecewise linear complex (PLC).
-Y	Preserves the input surface mesh (does not modify it).
-r	Reconstructs a previously generated mesh.
-q	Refines mesh (to improve mesh quality).
-R	Mesh coarsening (to reduce the mesh elements).
-A	Assigns attributes to tetrahedra in different regions.
-V	Verbose: Detailed information, more terminal output.
-h	Help: A brief instruction for using TetGen.

Table 3.1: *TetGen partial options and their descriptions.*

step, we optimize s_i and $\mathbf{R}_i$ with $\mathbf{p}'_i$ fixed. By contrast, $\mathbf{p}'_i$ are optimized with s_i and $\mathbf{R}_i$ fixed in global step.

Local step In this step, $\mathbf{p}'_i$ are fixed, and then we solve $\mathbf{R}_i$, s_i in sequence to construct *consistent* ASAP energy (3.7). For convenience, let us denote the edge $\mathbf{e}_{jk} := \mathbf{p}_j - \mathbf{p}_k$ and $\mathbf{e}'_{jk} := \mathbf{p}'_j - \mathbf{p}'_k$. Then we can change the formula (3.7) for cell i as

$$\sum_{(j,k)\in\mathcal{E}_i} w_{jk}\|\mathbf{e}'_{jk} - s_i\mathbf{R}_i\mathbf{e}_{jk}\|^2 + \alpha A \sum_{\mathcal{E}_l\in\mathcal{N}(\mathcal{E}_i)} w_{il}\|\mathbf{R}_i - \mathbf{R}_l\|_F^2. \tag{3.9}$$

First the optimal rotation $\mathbf{R}_i$ is optimized. Extending the equation (3.9) and dropping the terms that do not contain $\mathbf{R}_i$, we left

$$\begin{aligned} &\underset{\mathbf{R}_i}{\text{argmin}}\ Tr(-\mathbf{R}_i(2\sum_{(j,k)\in\mathcal{E}_i} s_i\mathbf{e}_{jk}\mathbf{e}'^{\,T}_{jk} + 2\alpha A \sum_{\mathcal{E}_l\in\mathcal{N}(\mathcal{E}_i)} w_{il}\mathbf{R}_l^T)) \\ &= \underset{\mathbf{R}_i}{\text{argmax}}\ Tr(\mathbf{R}_i\mathbf{S}_i), \end{aligned} \tag{3.10}$$

where Tr is the trace of a matrix, $\mathbf{S}_i$ is defined as

$$\mathbf{S}_i = 2\sum_{(j,k)\in\mathcal{E}_i} s_i\mathbf{e}_{jk}\mathbf{e}'^{\,T}_{jk} + 2\alpha A \sum_{\mathcal{E}_l\in\mathcal{N}(\mathcal{E}_i)} w_{il}\mathbf{R}_l^T.$$

The optimal rotation $\mathbf{R}_i$ is derived from the singular value decomposition of $\mathbf{S}_i = \mathbf{U}_i\mathbf{\Sigma}_i\mathbf{V}_i^T$:

$$\mathbf{R}_i = \mathbf{V}_i\mathbf{U}_i^T. \tag{3.11}$$

The determinate of a rotation matrix should be one, if $\det(\mathbf{R}_i) < 0$ then we change the sign of the column of $\mathbf{U}_i$ corresponding to the smallest singular.

Then the scale factor s_i is computed. Since the second term in (3.9) is independent with s_i, we only extend the first term and divide extended terms by s_i

$$\underset{s_i,\mathbf{R}_i}{\operatorname{argmin}} Tr(\sum_{(j,k)\in\mathcal{E}_i} w_{jk}(\frac{1}{s_i}\|\mathbf{e'}_{jk}\|^2 - 2\mathbf{R}_i\mathbf{e}_{jk}\mathbf{e'}_{jk}{}^T + s_i\|\mathbf{e}_{jk}\|^2)). \tag{3.12}$$

Taking derivative of (3.12) w.r.t. s_i and letting the derivative to be zero yields

$$s_i = \left(\frac{\sum_{(j,k)\in\mathcal{E}_i} w_{jk}\|\mathbf{e}'_{jk}\|^2}{\sum_{(j,k)\in\mathcal{E}_i} w_{jk}\|\mathbf{e}_{jk}\|^2}\right)^{\frac{1}{2}} \tag{3.13}$$

Global step In this step, vertex positions $\mathbf{p}'_i$ are optimized from $s_i, \mathbf{R}_i$ obtained by the local step.

Taking partial derivative of (3.7) w.r.t. the position $\mathbf{p}'_i$ (note that the second term has nothing to do with $\mathbf{p}'_i$), we arrive at

$$\begin{aligned}\frac{\partial E(\mathbf{p}')}{\partial \mathbf{p}'_i} =& 2\sum_{j\in\mathcal{N}(i)} (w_{ij}(3(\mathbf{p}'_i - \mathbf{p}'_j) - (s_i\mathbf{R}_i + s_j\mathbf{R}_j + s_m\mathbf{R}_m)(\mathbf{p}_i - \mathbf{p}_j)) \\ &+ w_{ji}(3(\mathbf{p}'_i - \mathbf{p}'_j) - (s_i\mathbf{R}_i + s_j\mathbf{R}_j + s_n\mathbf{R}_n)(\mathbf{p}_i - \mathbf{p}_j))),\end{aligned} \tag{3.14}$$

where $\mathcal{N}(i)$ is one-ring neighbors of vertex $\mathbf{p}'_i$; s_m, s_n and $\mathbf{R}_m, \mathbf{R}_n$ are the scaler factors and rotation matrices of the vertices $\mathbf{p}_m, \mathbf{p}_n$ which are the opposite vertices of the edge $\mathbf{e}_{ij}$. Setting partial derivative of (3.14) to zero gives the following sparse linear system of equations:

$$\sum_{j\in\mathcal{N}(i)}(w_{ij}+w_{ji})(\mathbf{p}'_i-\mathbf{p}'_j)=\frac{1}{3}\sum_{j\in\mathcal{N}(i)}(w_{ij}(s_i\mathbf{R}_i+s_j\mathbf{R}_j+s_m\mathbf{R}_m) +w_{ji}(s_i\mathbf{R}_i+s_j\mathbf{R}_j+s_n\mathbf{R}_n))(\mathbf{p}_i-\mathbf{p}_j). \quad (3.15)$$

Notice that the linear combination on the left-hand side is the discrete Laplace-Beltrami operator applied to $\mathbf{p}'$. Now the system of equations can be reduced as

$$\mathbf{L}\mathbf{p}'=\mathbf{d}, \quad (3.16)$$

where $\mathbf{L}$ represents the discrete Laplace-Beltrami operator, which only depends on the initial mesh, thus it can be pre-factored for efficiency; $\mathbf{d}$ is given by the right-hand side of (3.15).

Up to now, the optimization of CASAP energy can be summarised as Algorithm 3.1. Note that the order of the steps in the loop does not matter and have no effect on the results or convergence rate.

Algorithm 3.1 *Consistent* ASAP Energy Optimization

1: **while** not converged **do**
2: Compute $\mathbf{R}_i$ by solving equations (3.11).
3: Compute s_i by solving equations (3.13).
4: Compute $\mathbf{p}'$ and update surface $\mathcal{S}'$ by solving equation (3.16).
5: **end while**

3.5 Experiments

We first compare our *consistent as-similar-as-possible* (CASAP) deformation approach with other three deformation methods: ARAP [Sorkine & Alexa 2007], SR-ARAP [Levi & Gotsman 2015] and ASAP [Yamazaki et al. 2013]) in Figure 3.4. The iteration step and timing spent by methods are shown in Table 3.2. The result of ARAP is not satisfactory because its energy is not *consistent*. It lacks bending energy measurement, which makes surfaces bending freely without energy changed. This flaw can be obviously seen from bump in the plane example. SR-ARAP

overcomes the weakness of ARAP, offering a *consistent* energy. However, it does not allow local scalability. On the contrary, ASAP and CASAP are able to handle local scalability, which can be easily seen from Figure 3.5. ASAP allows piecewise scale but its energy is still not *consistent*. It has the same weakness as the ARAP, which may lead to undesirable results such as fold-over and self-intersection. CASAP combines the benefits of ASAP with the advantages of the SR-ARAP approach such that it can not only handle local scale but also guarantee the deformation smoothness. Moreover, in terms of isometric deformation, it produce competitive results as good as SR-ARAP, which can be observed from the bar twisting example.

	cylinder		bar		plane		armadillo	
	steps	time	steps	time	steps	time	steps	time
ARAP	1656	173.98	3775	351.04	301	19.08	1602	72.10
SR-ARAP	2364	289.99	5412	551.57	158	10.14	2086	98.23
ASAP	2966	429.42	1921	237.91	1087	62.02	1328	73.99
CASAP	1932	251.42	6887	928.52	237	13.91	3321	183.46

Table 3.2: *Iteration steps and timings (in seconds) in different deformation methods.*

We then compare CASAP, ASAP with the ground truth method VASAP in Figure 3.6 and evaluate the distance error to VASAP in Figure 3.7. Since CASAP makes up the bending measurement, it becomes consistent just like VASAP. On the contrary, ASAP can only measure the stretching deformation not the bending deformation, which may produce poor results, like the self-intersections shown in the bar twisting example. From this experiment, we could see our method produces closer results to VASAP, and has less errors compared with ASAP.

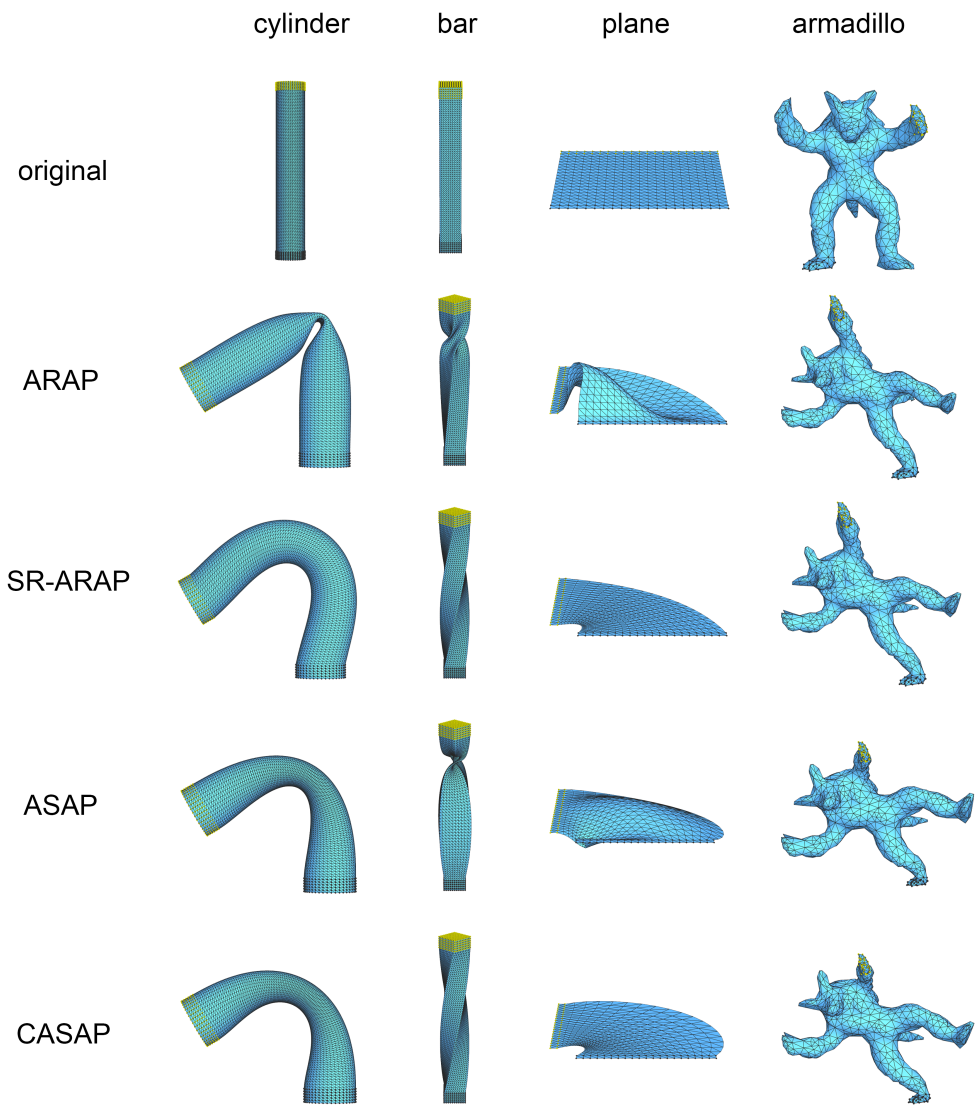


Figure 3.4: *Different deformation approaches comparison. Columns show different object transformations, while rows represent different deformation methods. The grey points are fixed and the yellow ones indicate control points.*

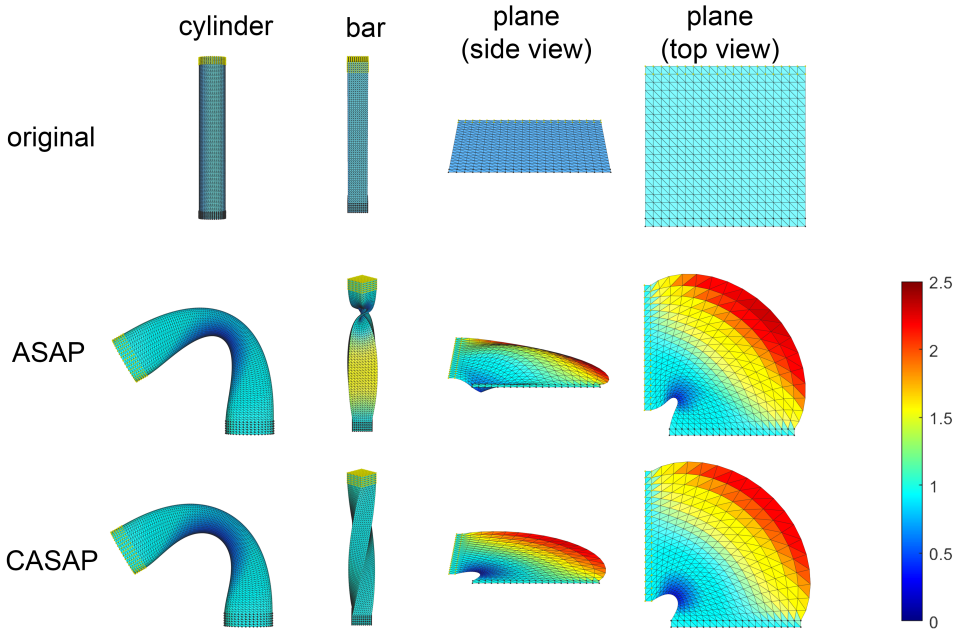


Figure 3.5: *The demonstration of scalability of ASAP and CASAP: Colder color means more shrinkage, while hotter color presents more expansion. The color bar indicates the value of the scale factor in each local cell.*

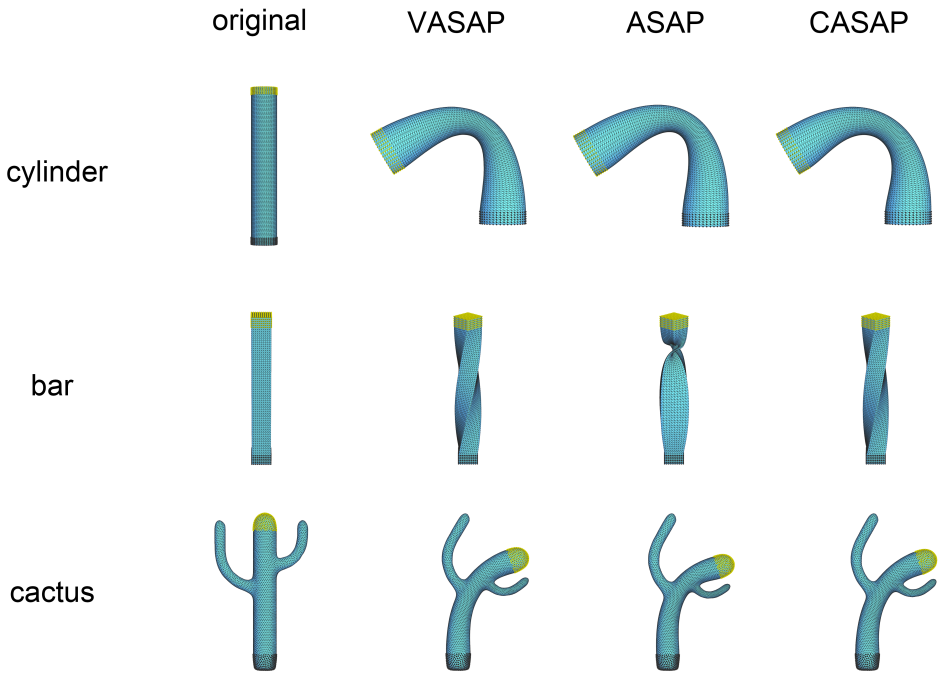


Figure 3.6: *Comparing ASAP and CASAP with the ground truth method (VASAP): Rows show different object transformations, while columns represent different deformation methods..*

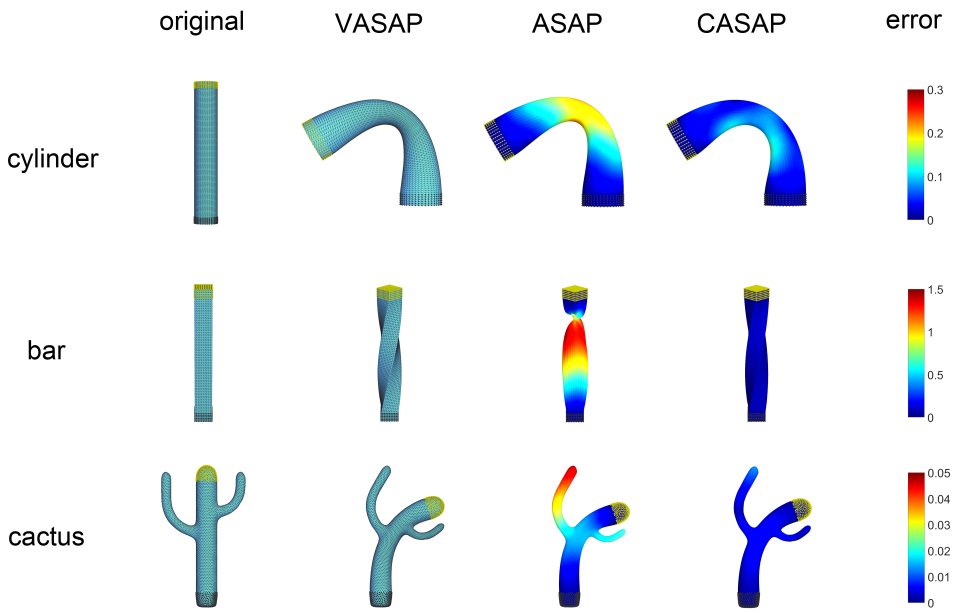
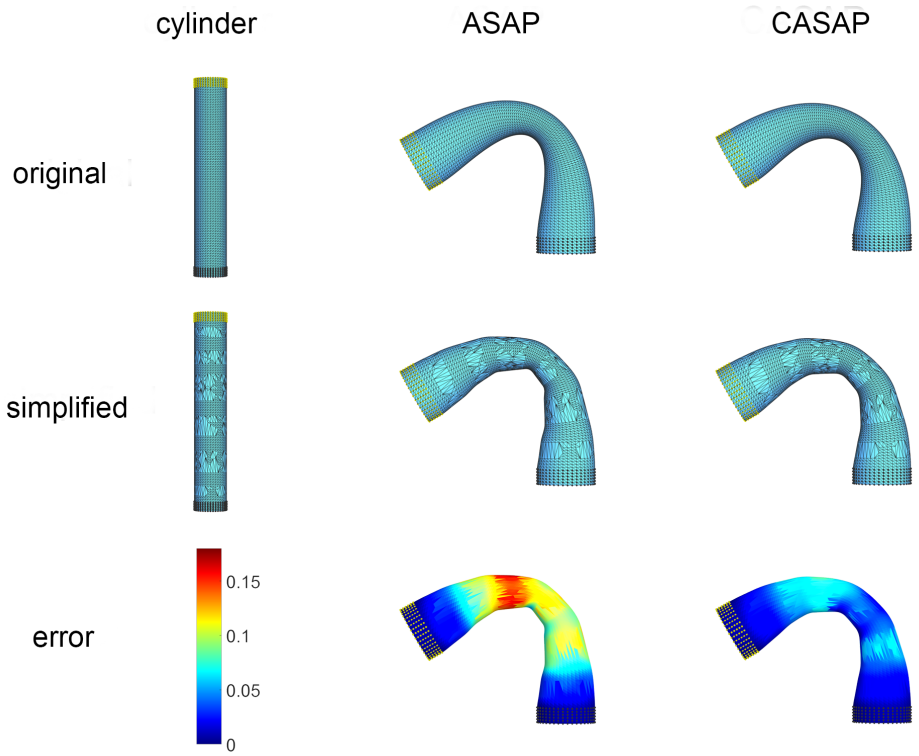
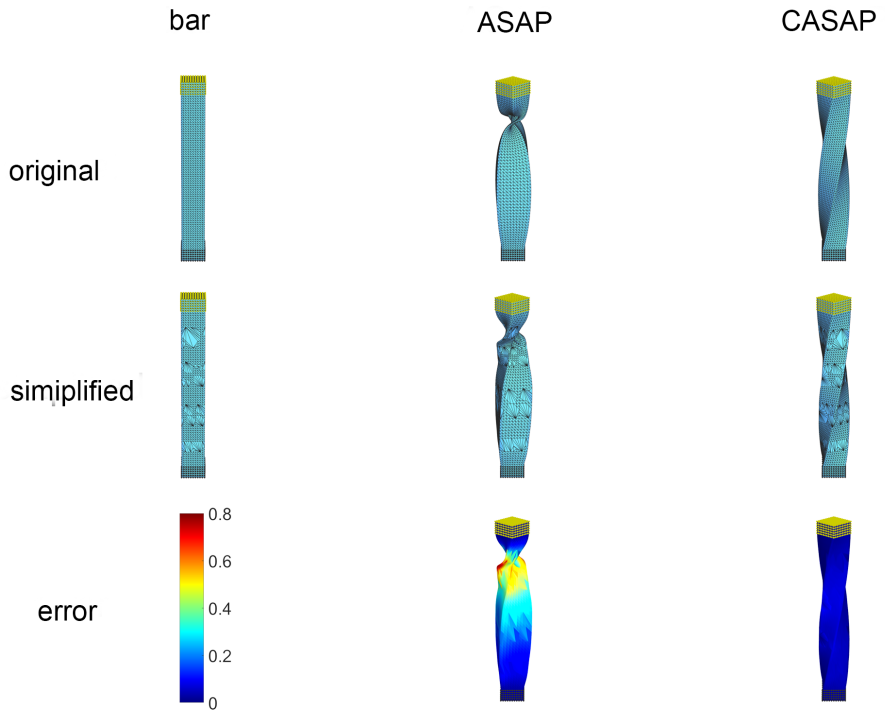


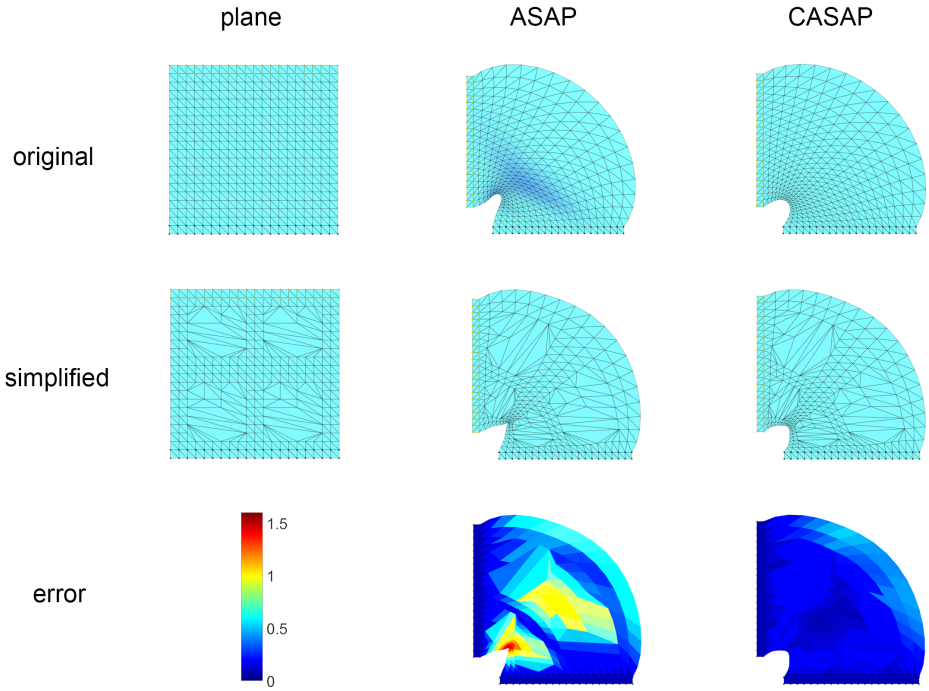
Figure 3.7: *The distance error of ASAP and CASAP to the ground truth method VASAP: Colder color means less error, while hotter color presents larger error. The color bar indicates the value of the distance error to VASAP.*



(a)



(b)



(c)

Figure 3.8: *Applying ASAP and CASAP on the same shape but with different discretization: (a) cylinder; (b) bar; (c) plane.*

To demonstrate the benefits of the consistent energy, we deform the same shapes but with different discretization and then evaluate the distance between them in Figure 3.8. From this experiment we could see that even though the discretization is different the CASAP deformed results are very close, this is benefited from our consistent energy.

3.6 Conclusion

We proposed a novel consistent as-similar-as-possible surface deformation method, which not only allows local scale to each discrete cell but also achieves the consistent discretization for surfaces. The important features of our approach are (1) robustness, resulting from the minimization procedure that is guaranteed to not increase energy in each step; (2) simplicity, as each iteration of the minimization solves a linear system; (3) efficiency, because the laplace system matrix is constant throughout the iterations and can be pre-factored just for once. We fitted the mapping differential into a similarity matrix, which is an isotropic scale factor times a rotation matrix. The scale factor is able to handle size difference while the rotation matrix part prevents local stretch and distortion. Meanwhile, the added rotation smoothing term compensates the bending energy which makes CASAP energy consistent, reducing the risk of fold-over and self-intersect occurrence. Our technique fills the missing gaps between SR-ARAP and ASAP. It combines the benefits of SR-ARAP and the advantage of ASAP, producing a consistent discretization and allowing local scalability to handle large deformation consistently without compromising the efficiency.

Chapter 4

Consistent As-Similar-As-Possible Non-isometric Surface Registration

Non-isometric surface registration, aiming to align two surfaces with different sizes and details, has been widely used in computer animation industry. Various existing non-isometric surface registration approaches have been proposed for accurate template fitting, nevertheless, two challenges remain. One is how to avoid the mesh distortion and fold-over of surfaces during transformation. The other is how to reduce the amount of landmarks that have to be specified manually. To tackle these challenges simultaneously, based on the deformation technique in the last chapter, we propose a *consistent as-similar-as-possible* (CASAP) surface registration approach. With a novel defined energy, it not only achieves the *consistent* discretization for the surfaces to produce accurate result, but also requires a small number of landmarks for correspondence with little user effort only. Besides, CASAP is constrained *as-similar-as-possible* so that angles of triangle meshes are preserved and local scales are allowed to change. Extensive experimental results have demonstrated the effectiveness of CASAP in comparison to the state-of-the-art approaches.

4.1 CASAP energy

In the following we denote by $\mathcal{S}$ a triangle mesh, whose topology is determined by n vertices and m triangles. Assuming we are deforming a mesh $\mathcal{S}$ into $\mathcal{S}'$ with the same connectivity as similar as possible, the piecewise linear geometric embedding of $\mathcal{S}$ is defined by the vertex positions $\mathbf{p}_i \in \mathbf{R}^3$, which is deformed into a different geometric embedding $\mathbf{p}'_i$.

Due to the advantages of CASAP deformation method introduced in the last chapter, we would like to deform the surface in a consistent as-similar-as-possible way as well. The CASAP energy is defined in equation (3.7). The importance of CASAP energy can be seen in Figure 4.1.

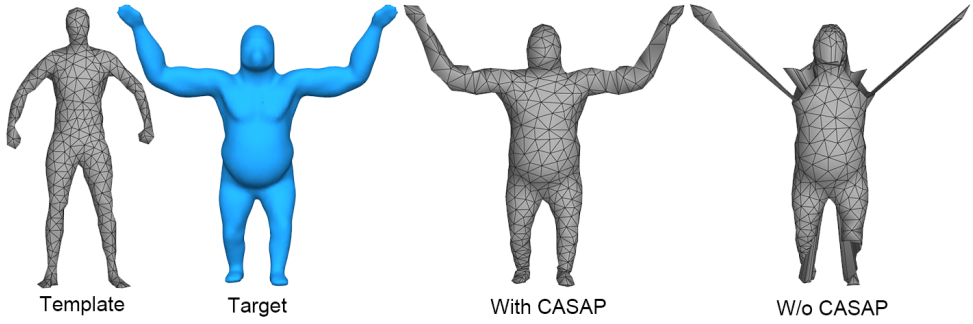


Figure 4.1: *Mid-scale fitting results with and without CASAP energy.*

4.2 Correspondence searching

Given a template surface $\mathcal{S}$ and a target one $\mathcal{T}$, the goal of surface registration is to deform the surface $\mathcal{S}$ into $\mathcal{S}'$ so that $\mathcal{S}'$ can be sufficiently close to surface $\mathcal{T}$ with quality preserved. In each iterative step, before deforming the template surface $\mathcal{S}'$, we need to find the correspondence between $\mathcal{S}'$ and $\mathcal{T}$. Many works [Yamazaki et al. 2013; Yoshiyasu et al. 2014; Gilles et al. 2010] regard the closest points as goal positions, however, correspondences chosen by these approaches are not quite appropriate as they only consider distances between the closest points of template and target surface. Inspired by [Papazov & Burschka 2011] we concern feature descriptors and smooth factor additionally. The novelty

is when we search the correspondence, instead of minimizing the feature distance between the template and the target directly, we minimize the difference between the feature distance and its 1-ring neighbor average. Starting from the closest points on the target, we then flood over their neighbours to find out the smallest matching energy points until converge. During the registration, the feature descriptor can be changed, however, starting from the closest point limits the searching range. The feature descriptor is only utilized to find the best similar within the restricted range. With the advance of iteration, the feature descriptor on the template will become similar to the target's as the template surface closing to the target. The matching energy E_m between points of the template and the target is defined as

$$E_m(\mathbf{p}_i, \mathbf{q}_j) = \|d_f(\mathbf{p}_i, \mathbf{q}_j) - \overline{d_f(\mathbf{p}_i, \mathbf{q}_j)}\|^2, \tag{4.1}$$

where $\mathbf{p}_i$ is vertex i on template surface and $\mathbf{q}_j$ is vertex j on target surface; the feature descriptors distance is defined as $d_f(\mathbf{p}_i, \mathbf{q}_j) = f(\mathbf{p}_i) - f(\mathbf{q}_j)$, where $f(v)$ is the feature vector for vertex v, we concatenate all feature descriptors into a single feature vector; the mean value distance $\overline{d_f(\mathbf{p}_i, \mathbf{q}_j)} = \frac{1}{|\mathcal{N}(j)|+1} \sum_{k \in \mathcal{N}(j) \cup j} d_f(\mathbf{p}_i, \mathbf{q}_k)$, where $\mathcal{N}(j)$ is the 1-ring neighbours of vertex j on the target surface.

There is a great number of feature descriptors that characterize the geometric properties of the point or of its neighbourhood, often in a multi-scale way, for example, various notions of curvature (Gaussian, mean) [Meyer et al. 2003], diffusion-based descriptors, such as the Heat or Wave Kernel Signatures [Sun et al. 2009; Aubry et al. 2011], or more classical descriptors such as spin images or shape contexts [Johnson & Hebert 1999; Belongie et al. 2002]. In our experiment we concatenate vertex position, vertex normal, multi-scale mean curvatures [Panozzo et al. 2010], Wave Kernel Signatures [Aubry et al. 2011] and Scale-invariant Heat Kernel Signatures [Bronstein & Kokkinos 2010] to form a feature vector.

In order to prevent unnecessary matchings, we filter out the pairs if the distance between them exceeds D or if the angle between their

normals exceeds a threshold Θ. Thus the algorithm of finding correspondence $\mathbf{q}_{\text{idx}(i)}$ on the target surface for each point on the template can be summarized as Algorithm 4.1, where idx(i) is the index of the target point that is matched with template vertex i. After given the cor-

Algorithm 4.1 Find correspondence for template vertex p_i

```
Find the closest point q_j on the target
if the distance between p_i and q_j exceeds D or the angle between their normals exceeds Θ then
    return NULL
end if
k = j
do
    j = k
    Find k ∈ N(j) ∪ j which minimizes E_m
while k ≠ j
idx(i) = k
return q_idx(i)
```

respondence of template vertices, the template surface can be attracted towards the target according to the matching pairs. However, in order to avoid extreme distortion in tangential space, rather than attracting the template points to their correspondences directly, we attract them to the projections of their correspondences on their normals denoted by $Proj(\mathbf{q}_{\text{idx}(i)})$ (Figure 4.2). Now the correspondence constraint energy can be expressed as

$$E_c(\mathbf{p}') = \|\mathbf{C}_c\mathbf{p}' - Proj(\mathbf{D}_c\mathbf{q})\|_F^2, \tag{4.2}$$

where $\mathbf{p}', \mathbf{q}$ are the vertex positions on surface $\mathcal{S}', \mathcal{T}$ respectively, and $\mathbf{C}_c, \mathbf{D}_c$ are the sparse matrices that define the filtered matching correspondences between $\mathcal{S}'$ and $\mathcal{T}$. Assuming the m-th correspondence is $\mathbf{p}_i$ on $\mathcal{S}'$ and $\mathbf{q}_{\text{idx}(i)}$ on $\mathcal{T}$, then

$$\mathbf{C}_c(m,n) = \begin{cases} 1, & \text{if } n = i \\ 0, & \text{if } n \neq i \end{cases}, \mathbf{D}_c(m,n) = \begin{cases} 1, & \text{if } n = \text{idx}(i) \\ 0, & \text{if } n \neq \text{idx}(i) \end{cases}.$$

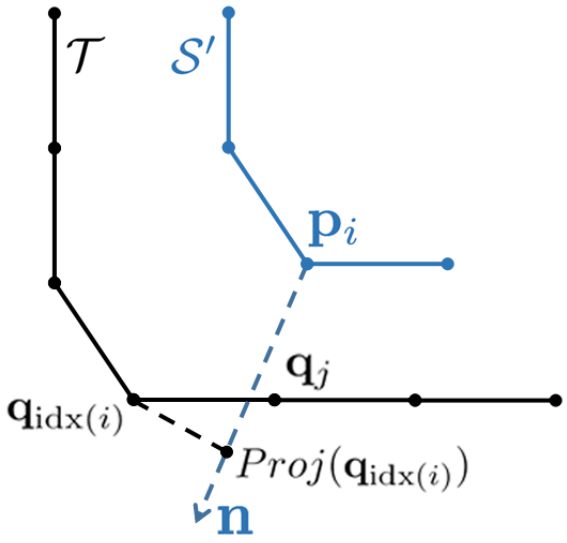


Figure 4.2: $\mathbf{q}_j$ *is the closest vertex on the target to* $\mathbf{p}_i$*;* $\mathbf{q}_{idx(i)}$ *is the correspondent vertex found by minimizing the matching energy;* $\mathbf{n}$ *is the normal vector of* $\mathbf{p}_i$*;* $Proj(\mathbf{q}_{idx(i)})$ *is the projection of* $\mathbf{q}_{idx(i)}$ *onto normal vector* $\mathbf{n}$*.*

4.3 CASAP surface registration

Based on the above two constraints, we propose our novel consistent as-similar-as-possible surface registration method. Let $\mathbf{p}, \mathbf{p}', \mathbf{q}$ denote the vertex positions on surface $\mathcal{S}, \mathcal{S}', \mathcal{T}$ respectively, we define the total cost function as

$$E(\mathbf{p}') = w_d E_d(\mathbf{p}') + w_c E_c(\mathbf{p}') + w_f E_f(\mathbf{p}'), \tag{4.3}$$

where E_d constrains deformation ASAP consistently, E_c penalizes distances between the points of template and their correspondences on the target, and E_f penalizes distances between the feature points of template and target surface. The weights before these energy terms adjust the influence they account for in total energy. As E_d, E_c have been introduced before, here we only introduce the feature point constraint energy E_f.

4.3.1 Feature Point Constraints

For the fitting of the template's pose and size to the target, several feature correspondences are required to established. Feature point constrains are

designed to drag feature points on the template towards corresponding target ones. This constraint energy can be represented as

$$E_f(\mathbf{p}') = \|\mathbf{C}_f\mathbf{p}' - \mathbf{D}_f\mathbf{q}\|_F^2, \tag{4.4}$$

where $\mathbf{C}_f, \mathbf{D}_f$ are the sparse matrices that define the feature point pairs between $\mathcal{S}'$ and $\mathcal{T}$.

4.3.2 Optimization

In this subsection, we introduce the optimization algorithm to minimize the total energy in (4.3). There are two loops in the optimization: the outer loop searches for the correspondent vertices to construct E_c, the inner loop optimizes the deformed vertex positions by minimizing $E(\mathbf{p}')$. Once the inner loop is converged, weights are adjusted and a new outer iteration starts again. Note that in the inner loop except the vertex positions $\mathbf{p}_i$ are unknown, s_i and $\mathbf{R}_i$ in (3.7) are also unknown for each vertex. We employ the alternating optimization scheme following [Sorkine & Alexa 2007; Yamazaki et al. 2013; Levi & Gotsman 2015] to solve them respectively. Each inner iteration consists of a local step followed by a global step. In local step, we optimize s_i and $\mathbf{R}_i$ with $\mathbf{p}'_i$ fixed. By contrast, $\mathbf{p}'_i$ are optimized with s_i and $\mathbf{R}_i$ fixed in global step. The local step and the global step are similar to the optimization of CASAP deformation method in the last chapter except the total energy optimization. Taking derivative of the total energy (4.3) w.r.t. $\mathbf{p}'$ gives us a linear system:

$$\mathbf{A}^T\mathbf{A}\mathbf{p}' = \mathbf{A}^T\mathbf{b}, \tag{4.5}$$

where

$$\mathbf{A} = \begin{pmatrix} w_d\mathbf{L} \\ w_c\mathbf{C}_c \\ w_f\mathbf{C}_f \end{pmatrix}, \mathbf{b} = \begin{pmatrix} w_d\mathbf{d} \\ w_c Proj(\mathbf{D}_c\mathbf{q}) \\ w_f\mathbf{D}_f\mathbf{q} \end{pmatrix}.$$

Up to now, the routine of *consistent* ASAP surface registration can be summarised as Algorithm 4.2.

Algorithm 4.2 *Consistent* ASAP Surface registration

```
Specify the feature points.
while not converged do
    Adjust weights in (4.5) and construct E_c
    while not converged do
        Compute R_i by solving equations (3.11).
        Compute s_i by solving equations (3.13).
        Compute p' and update surface S' by solving equation (4.5).
    end while
end while
```

4.3.3 Fitting steps

In this subsection, we discuss the details of fitting the template. To improve the efficiency and robustness of registration, we take a coarse-to-fine fitting strategy [Yoshiyasu et al. 2014]. Instead of fitting overall template surface from the beginning, a coarse mesh is extracted from the original template mesh and then fitted to several feature points to roughly adjust the overall size of the template. In this way, approximated goal positions are obtained which is a better initial guess of fine fitting leading to fast converge and it also reduces the fold-over occurrence. Afterwards, a dense mesh is rebuilt from the deformed coarse mesh and fine fitting step is performed to produce the final result. The whole registration overview can be found in Figure 4.3.

Specifically, there are four fitting steps through the whole registration process: initialization, coarse fitting, mid-scale fitting and fine fitting:

Initialization In this step, a coarse mesh is extracted from the template first. We employ the farthest point sampling approach [Moenning & Dodgson 2003] to sample certain number of vertices to represent the shape of objects approximately (Figures 4.3a, 4.4). Note that all of the sampled vertices are the subset of the original vertex set. Then the geodesic remeshing technique [Peyré & Cohen 2006] is used to generate the coarse mesh from the sampled points (Figure 4.3b).

Coarse Fitting We utilize the similarity constraints E_d and feature

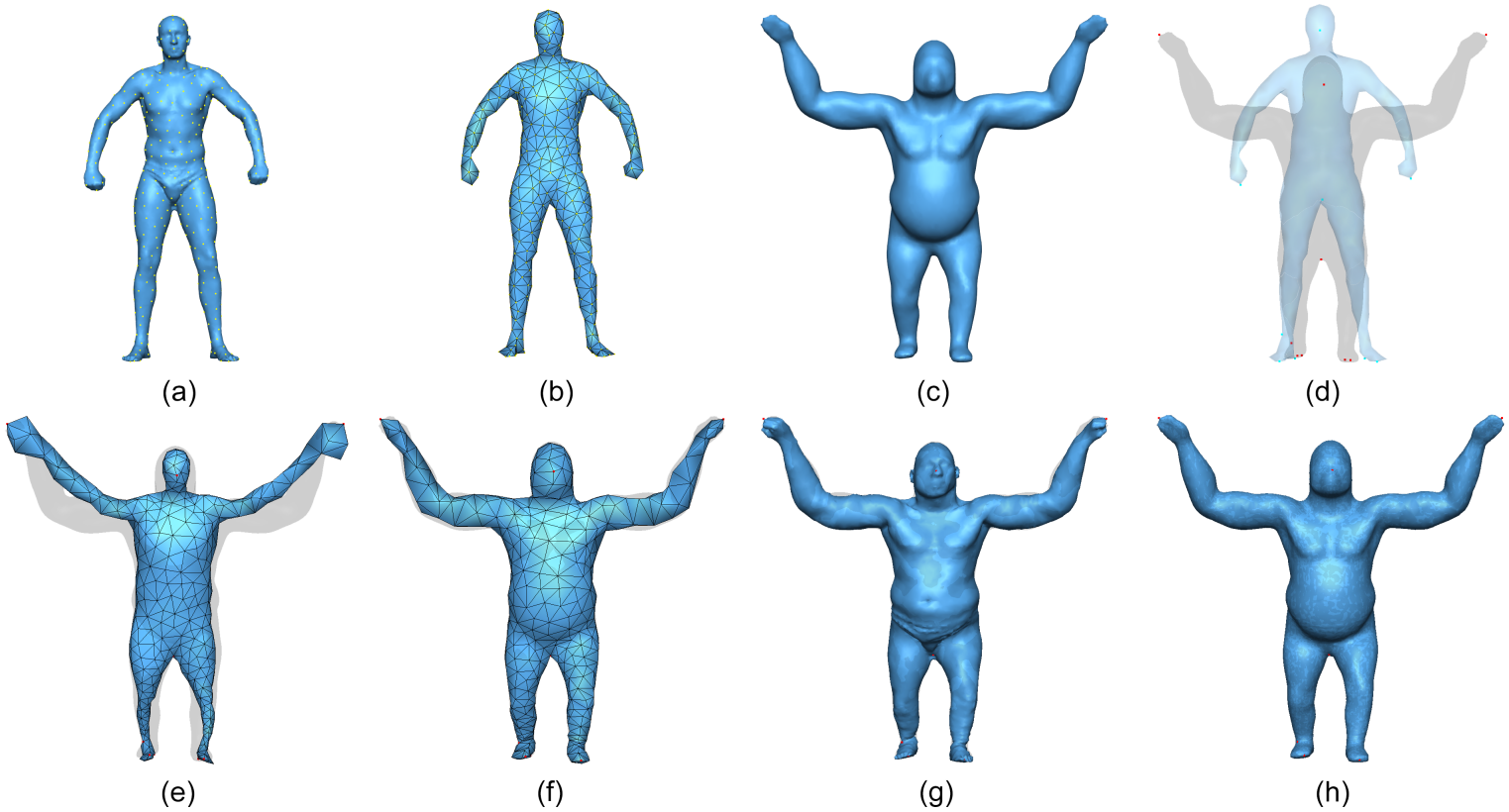


Figure 4.3: *Surface registration algorithm overview: (a) Sampled points (marked as yellow dots) via farthest point sampling technique; (b) Remeshing from the sampled points as embedded coarse mesh; (c) Input of target surface; (d) The feature points specified by users (red dots for target and cyan dots for template); (e) Coarse fitting; (f) Mid-scale fitting; (g) Reconstructed through embedded deformation; (h) Fine fitting.*

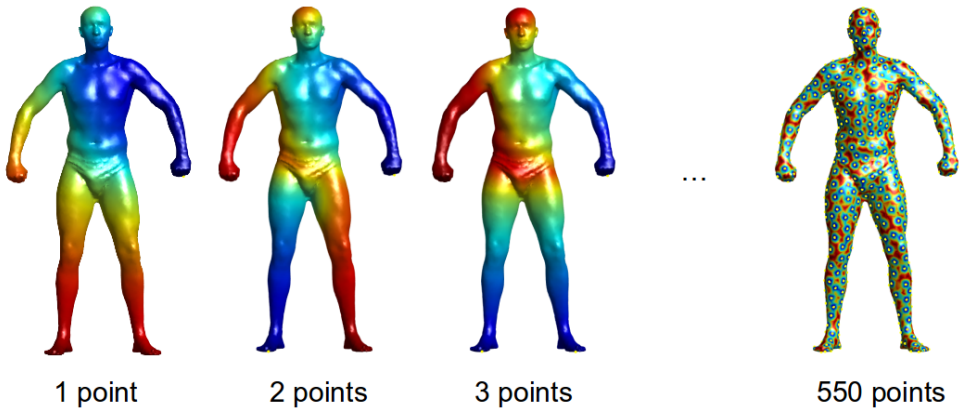


Figure 4.4: *The farthest point sampling approach is applied to sample certain amount of vertices on the surface. The colder color means smaller geodesic distance to sampled vertex set, while the hotter color means larger distance. The first sampled point can be appointed or chosen randomly.*

point constraints E_f to fit the coarse mesh to several feature points on the target so that the size and pose of the template are roughly adjusted to the target (Figure 4.3e). Note that this step will change the scale of the template to roughly align with the target, so no global scaling is required to match the size of the template to the target. The energy function is expressed as:

$$E(\mathbf{p}') = w_d E_d(\mathbf{p}') + w_f E_f(\mathbf{p}'), \tag{4.6}$$

Mid-scale Fitting After fitting the template roughly to the target using feature points, the coarse mesh is deformed gradually toward the target. Apart from the two constraints adopted in the first step, correspondence constraints are also applied to achieve template attraction (Figure 4.3f).

Fine Fitting In this stage, a dense mesh is first reconstructed from the deformed coarse mesh by embedded deformation [Sumner et al. 2007] (Figure 4.3g). The extracted coarse mesh is considered as deformation graph laid under the the dense mesh. From formulas (3.13) and (3.11), we associate an affine transformation with each vertex in the coarse graph.

The deformed positions of vertices in the dense mesh can be calculated from the transformations of the deformation graph. We use the same approach as [Yoshiyasu et al. 2014] to rebuild the dense mesh. Again, all the constraints are performed to fit the dense mesh to the target (Figure 4.3h).

4.3.4 Weights and parameters

In the initialization step, we regard the feature point constraints as boundary condition to induce deformation. In next two steps, we set $D = 0.02r_{box}$ and $\Theta = 90°$, where r_{box} is the bounding box diagonal. As for the weights in the linear system (4.5), we use $w_d = 1000$, $w_c = 5$, $w_f = 10^5$ in the coarse fitting stage and divide w_d by 1.1 after every iteration until it less than 1. In the fine fitting, we take the same procedure with $w_f = 1$. During the iteration all the parameters are fixed except w_d.

4.3.5 Experiments and results

Generic models We apply CASAP registration technique to register from one human head with holes to a face scanning from another human (Figure 4.5); from a human body to a gorilla (Figures 4.5, 4.3); from a pig to a horse (Figure 4.6). Each pair has large difference on size or details. CASAP not only is able to handle size difference as shown in whole-body registration example in Figure 4.3, but also can capture geometrical details such as the human expression (Figure 4.5) and preserve the connectivity of the template well, thus reducing the risk of producing fold-over (Figures 4.5, 4.6).

We then compare our registration technique to other state-of-the-art algorithms: *as-conformal-as-possible* surface registration (ACAP) [Yoshiyasu et al. 2014], similarity-invariant shape registration (ASAP) [Yamazaki et al. 2013], the embedded deformation technique (ED) [Sumner et al. 2007], the shape matching based registration technique that minimizes the *as-similar-as-possible* energy (SM-ASAP) [Papazov & Burschka 2011],

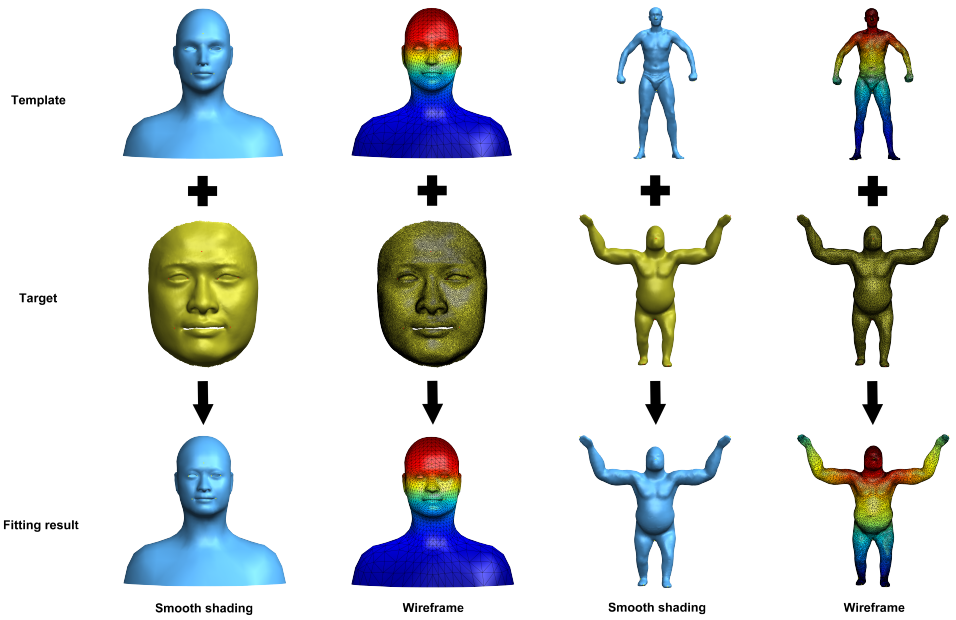


Figure 4.5: Consistent as-similar-as-possible *(CASAP) non-isometric registration. Given a small number of feature correspondences (seven for the head registration and nine for the whole-body registration) only, CASAP not only is capable of fitting the template towards the target with different size (revealed in the whole-body registration example), but also captures the details well (shown in the face registration example) and preserves the structure of the template (seen from the colored wireframe shading mode).*

the Laplacian surface editing technique (LSE) [Sorkine et al. 2004] and the registration technique that utilizes the point-based deformation smoothness regularization (PDS) [Amberg et al. 2007] in Figure 4.6. All these methods have the same input of feature points except ASAP and SM-ASAP, which do not require specifying feature points but are only able to handle surfaces with close initial alignment and similar poses. ASAP and ACAP are basically equivalent. The only difference is they have different formula leading to different optimization technique. ACAP employs nonlinear conformal stiffness and regularization terms in registration process, which produces the closest results to CASAP. However, since the regularization energy it adopts is not *consistent*, fold-overs still occur around the left wrist of gorilla and the neck of horse. ED is an isometric counterpart of ACAP. As it cannot adjust local scale, ED may produce poor initial shape estimation, which makes parts of surface converge to inaccurate places as shown at the right leg of gorilla. LSE cannot handle

large deformation as it use a linear approximation of similar transformation. PDS is based on smoothness regularization, but it is too weak to against shear distortions. Only CASAP exhibits no fold-over and almost no distortion in the examples, which produce quite pleasant visual results.

From the perspective of quantitative evaluation, following the same criterion as in [Yoshiyasu et al. 2014], we measure 1) distance error, which is the average distance from the vertices of the deformed template to the corresponding points of the target relative to the bounding box diagonal, 2) angle error, which is the average angle deviation from the template, 3) bending error, which is the average deviation in dihedral angles from the template, 4) intersection error, which is the number of self-intersecting faces. These statistics can be found in Figure 4.7. All the errors of CASAP are the smallest among all the techniques except the bending error in horse example, which is because of the LSE's disability of handling large rotations. The number of self-intersecting faces is zero, which reveals the ability of CASAP to reduce the change of fold-over and shear distortion appearance.

The number of iteration steps and timings are shown in Table 4.1. The time required for a single inner iteration of CASAP is minimum. Although it requires more iteration steps than ACAP to converge, the total registration time it spends is less than ACAP.

Number of feature points required Previous method [Yeh et al. 2011] requires specifying 20-70 feature points (34 for registration from camel to horse, 21 for registration from old man head to Venus head), whereas our technique requires less than 20 points: 7 for the face registration (Figure 4.5), 9 for the whole-body registration(Figure 4.5, 4.3), 15 for registration from pig to horse (Figure 4.6). Theoretically, the more feature points specified, the better result is. However, in our experience, only a small amount of feature points are required as long as the coarse

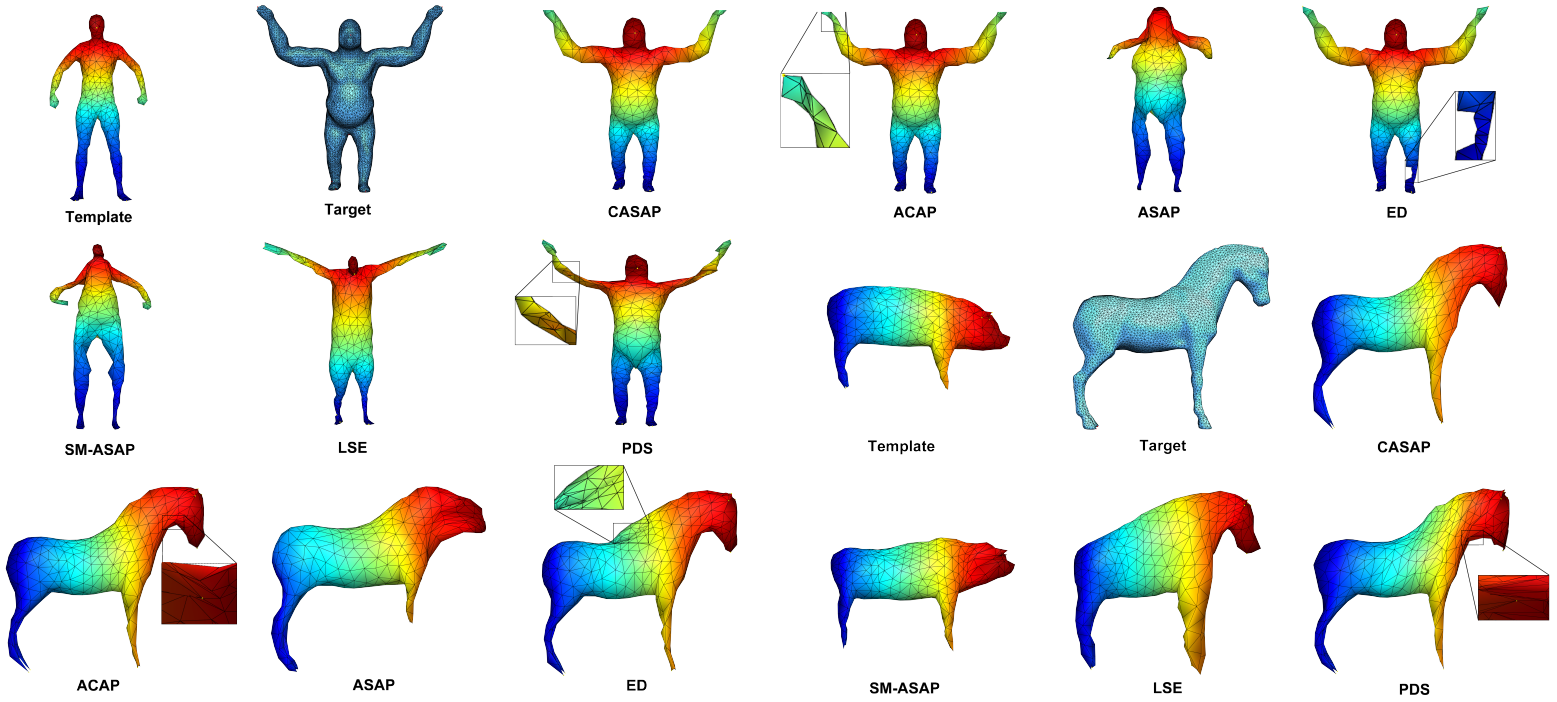


Figure 4.6: *Different surface registration methods comparison. The left two columns are inputs while the rest are outputs by different surface registration methods. The yellow and red dots indicate the feature points on the template and the target respectively. The corresponding points are with same colors.*

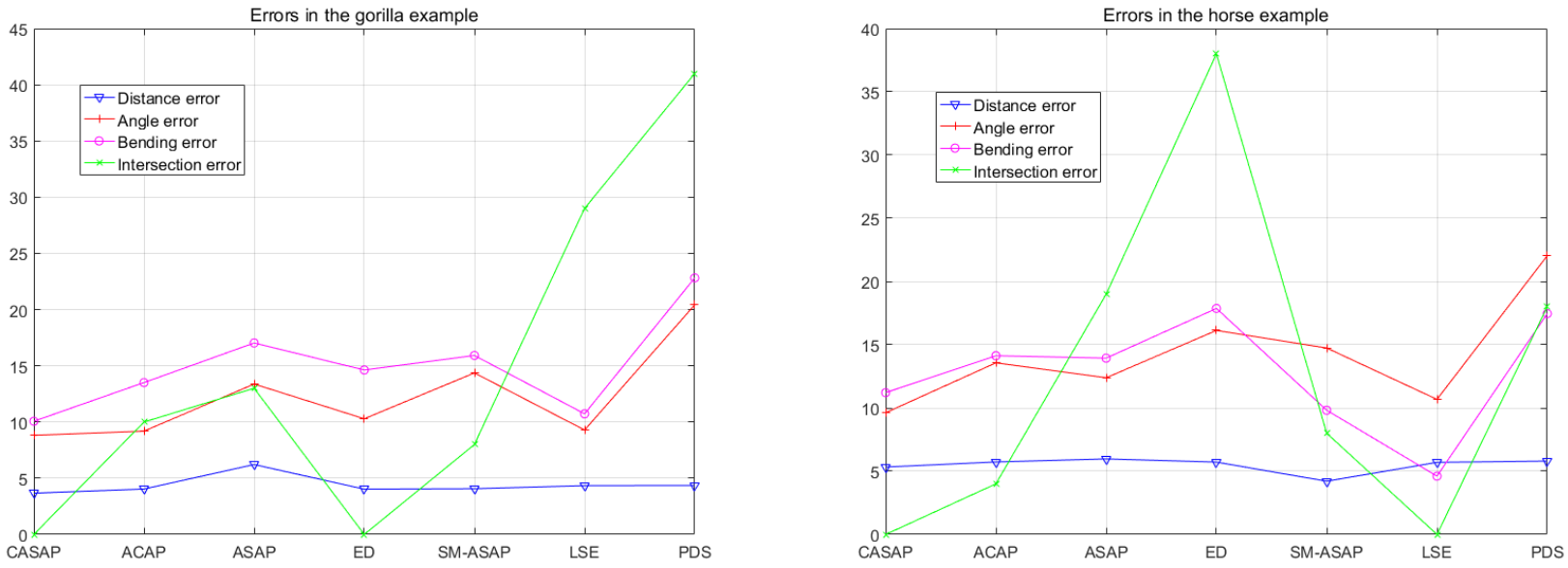
template can roughly aligned with the target. That is because CASAP provides a good initial shape approximation, and the *consistent* energy preserves the template structure and angles well.

Limitation Although our *consistent* ASAP deformation technique and the coarse-to-fine strategy can efficiently reduce the chance of fold-over, it cannot solve this issue entirely, especially for model with large curvature. An easy solution is to add more feature points around the fold-overs and adjust the position of them to achieve better result. Other methods such as fold-over removing technique [Yeh et al. 2011] or bounded distortion mapping [Lipman 2012] can also be utilized to solve this issue.

Another limitation is that we can not achieve automatic registration with no user input. Therefore, the quality of the feature points specified by users will directly influence the registration result.

4.4 Summary

In this chapter, we have presented a novel surface registration approach (CASAP) that constrains deformations locally as similar as possible. With the proposed *consistent* regularization energy, CASAP not only results in *consistent* discretization for surface but also reduces the occurrence of fold-over and shear distortion. Experiments have shown that CASAP produced more accurate fitting results and preserved angles better than previous methods.



(a) **(b)**

Figure 4.7: *Quantitative comparison for the gorilla and horse registration results respectively.*

	CASAP				ACAP				ASAP				ED				SM-ASAP				LSE				PDS			
	#O	#I	Inner	Total	#O	#I	Inner	Total	#O	#I	Inner	Total	#O	#I	Inner	Total	#O	#I	Inner	Total	#O	#I	Inner	Total	#O	#I	Inner	Total
Gorilla	54	2882	**0.035**	100.87	73	313	0.338	105.94	20	683	0.038	25.95	10	1547	0.379	586.91	9	50	0.376	18.783	**1**	**1**	0.357	**0.357**	500	2166	0.036	77.976
Horse	54	2915	**0.032**	93.28	73	276	0.423	116.62	20	1501	0.033	49.53	10	1597	0.508	811.95	9	48	0.355	17.025	**1**	**1**	0.371	**0.371**	500	1870	0.035	65.45

Table 4.1: *Iteration steps and timings (in seconds). #O, #I indicate the number of outer iteration steps and total inner iteration steps respectively. "Inner" indicate the average time required for each inner iteration step. "Total" indicate the total registration time.*

Chapter 5

Huber-L_1 Based Non-isometric Surface Registration

Although the CASAP surface registration methods can produce more accurate results than the state-of-the-arts, it assumes the target is clean and has no noise, which is hard to achieve, especially for the commodity depth sensors like Microsoft Kinect and ASUS XTion. Therefore, a surface registration which is robust against noise is highly desirable. In this chapter, we propose a Huber-L_1 based non-isometric surface registration and solve it by the alternating direction method of multipliers. With a Huber-L_1 regularized model constrained on the transformation variation and position difference, our method is robust to noise and produces piecewise smooth results while still preserving fine details on the target. The introduced as-similar-as-possible energy is able to handle different size of shapes with little stretching distortion. Extensive experimental results have demonstrated that our method is more accurate and robust to noise in comparison with the state-of-the-arts.

5.1 Surface Registration

We adopt a coarse-to-fine fitting strategy to implement the whole registration process in three steps: coarse fitting, mid-scale fitting and fine fitting. For each step, different energy terms are used and combined, which will be introduced at their first appearance. The comparison results with and without some energy terms will be illustrated to stress the significance of these energies.

5.1.1 Notations

Suppose the template mesh is composed of n vertices $\mathcal{P} \triangleq \{\mathbf{p}_1, \cdots, \mathbf{p}_n\}$, where $\mathbf{p}_i \triangleq [x_i, y_i, z_i]^\top$ is a 3D vertex position in Euclidean coordinate. In coarse fitting step, n is the vertex number of coarse template mesh. The vertices of the target are denoted as $\mathcal{Q}$=$\{\mathbf{q}_1,\cdots,\mathbf{q}_m\}$. For non-rigid registration, a 3×4 affine transformation matrix $\mathbf{A}_i \triangleq [\mathbf{X}_i, \mathbf{t}_i]$ is associated with each vertex $\mathbf{p}_i$ of the template, where $\mathbf{X}_i$ is a 3×3 linear transformation matrix and $\mathbf{t}_i$ is a 3×1 translation vector. For simplification, we concatenate $\mathbf{p}_i, \mathbf{q}_i, \mathbf{X}_i, \mathbf{t}_i$ into a $n \times 3$ matrix $\mathbf{P} \triangleq [\mathbf{p}_1 \cdots \mathbf{p}_n]^\top$, a $m \times 3$ matrix $\mathbf{Q} \triangleq [\mathbf{q}_1 \cdots \mathbf{q}_m]^\top$, a $3n \times 3$ matrix $\mathbf{X} \triangleq [\mathbf{X}_1 \cdots \mathbf{X}_n]^\top$ and a $n \times 3$ matrix $\mathbf{T} \triangleq [\mathbf{t}_1 \cdots \mathbf{t}_n]^\top$ respectively. Similarly, the vertices on the template dual mesh $\mathcal{P}^*$ are denoted by $\mathbf{P}^* \triangleq [\mathbf{p}_1^* \cdots \mathbf{p}_{n^*}^*]^\top$, where n^* is the number of the vertices on the template dual mesh, which is also equal to the number of triangle faces on the template primal mesh. Again, a translation vector will be assigned to each dual vertex, all of which can be concatenated as $\mathbf{T}^* \triangleq [\mathbf{t}_1^* \cdots \mathbf{t}_{n^*}^*]^\top$.

5.1.2 Coarse fitting

Instead of fitting the fine template surface from the beginning, a coarse mesh extracted from the origin template mesh is used to fit for efficiency, as the coarse mesh involves less unknown parameters. We employ the farthest point sampling approach [Moenning & Dodgson 2003] to sample certain number of vertices to approximately represent the shape of the template (Figure 5.1a). Note that all the sampled vertices are the subset of the original vertex set. The geodesic remeshing technique [Peyré

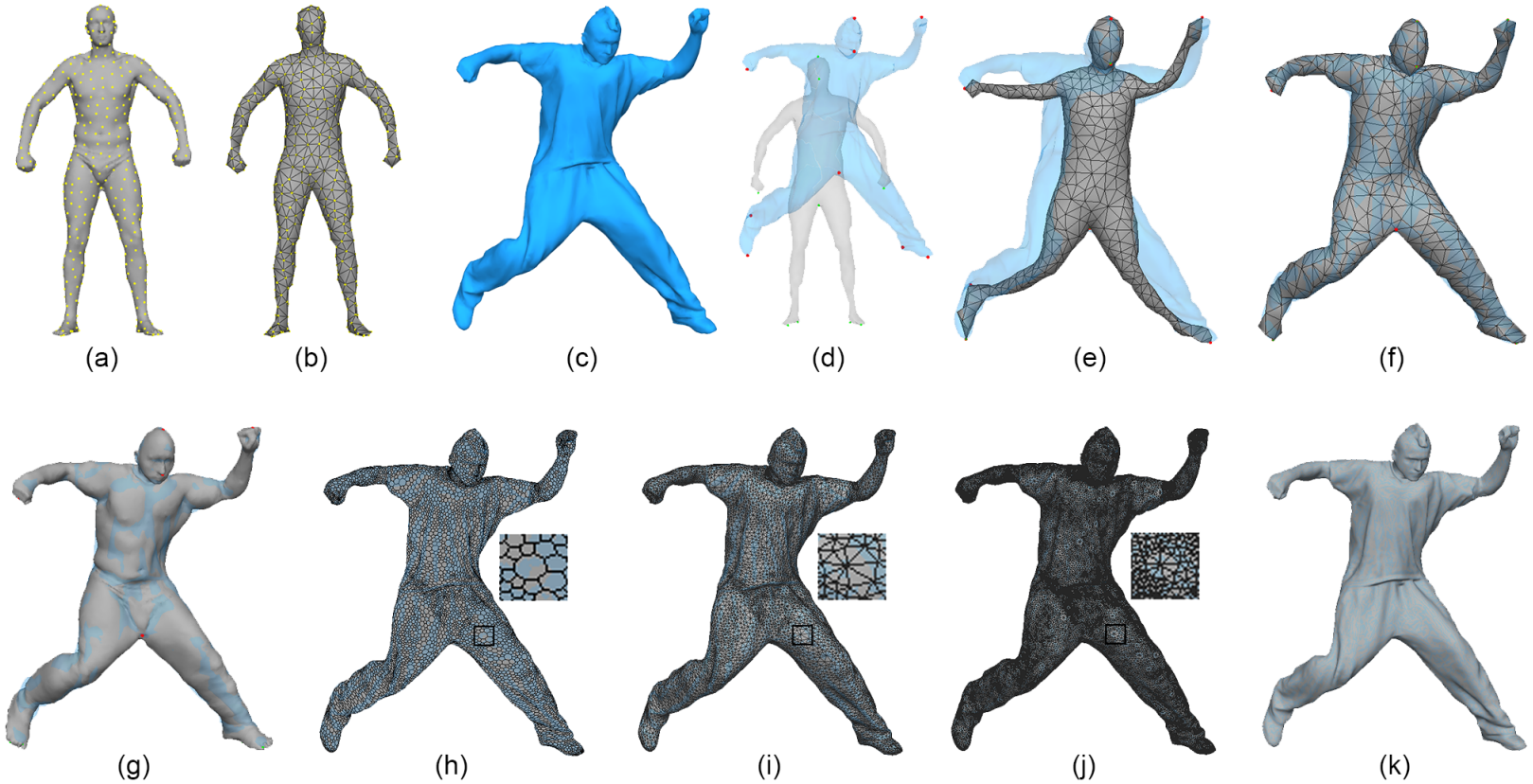


Figure 5.1: *Huber-L_1 algorithm: (a) Sampled points (yellow dots); (b) Remeshing from the sampled points as embedded coarse mesh; (c) Input of target surface; (d) 9 feature points specified by user (red dots for target and green dots for template); (e) Coarse fitting; (f) Mid-scale fitting; (g) Reconstructed through embedded deformation; (h) Fine fitting on the dual domain; (i) Fine fitting on the primal domain; (j) Subdivision; (k) The final result.*

& Cohen 2006] is then applied to generate the coarse mesh out of the samples vertices (Figure 5.1b). Afterwards, the feature points between the coarse template mesh and the target are specified by users (Figure 5.1d). The coarse template is then fitted to the specified feature points to approximate the overall size of the target (Figure 5.1e). The total energy in coarse fitting step is composed of four energies: $\mathrm{E_f}$ penalizes the distances between the feature points of template and target surface; $\mathrm{E_{ASAP}}$ constrains deformation ASAP; $\mathrm{E_r}$ penalizes the transformation variation; $\mathrm{E_l}$ penalizes the edge lengths difference locally:

$$\mathrm{E_{coarse}}(\mathbf{X}, \mathbf{T}) = w_{\mathrm{f}}\mathrm{E_f} + w_{\mathrm{ASAP}}\mathrm{E_{ASAP}} + w_{\mathrm{r}}\mathrm{E_r} + w_{\mathrm{l}}\mathrm{E_l}. \tag{5.1}$$

Each energy term will be introduced as follows, and w_{f}, w_{ASAP}, w_{r} and w_{l} are the weights which represent the influence of each energy term.

Feature point constraint

To pull feature points on the template towards their correspondence on the target, we define the feature point constraint energy as:

$$\mathrm{E_f}(\mathbf{T}) = \frac{1}{2}\sum_{i\in\mathcal{F}} \|\mathbf{p}_i + \mathbf{t}_i - \mathbf{q}_{\mathrm{idx}(i)}\|_F^2,$$

where $\mathcal{F}$ is the template index set of the feature points, $\mathbf{p}_i$ is the position of i-th feature point on the template, $\mathrm{idx}(i)$ is the index of the corresponding feature point on the target. To fit this energy (and other energies subsequently) into ADMM [Boyd et al. 2011] for optimization, we need to rewrite the energy (and other energies subsequently introduced) in matrix form. We define two sparse matrices $\mathbf{C}_{\mathrm{f}}$, $\mathbf{D}_{\mathrm{f}}$ which select the feature point pairs between the template and the target. Assuming the r-th feature point pair is $\mathbf{p}_i$ on the template and $\mathbf{q}_{\mathrm{idx}(i)}$ on the target, then

$$\mathbf{C}_{\mathrm{f}}(r,s) = \begin{cases} 1, & \text{if } s = i \\ 0, & \text{if } s \neq i \end{cases}, \quad \mathbf{D}_{\mathrm{f}}(r,t) = \begin{cases} 1, & \text{if } t = \mathrm{idx}(i) \\ 0, & \text{if } t \neq \mathrm{idx}(i) \end{cases}.$$

Therefore, we can rewrite $\mathrm{E_f}$ as

$$\mathrm{E_f}(\mathbf{T}) = \frac{1}{2}\|\mathbf{C}_\mathrm{f}(\mathbf{P}+\mathbf{T}) - \mathbf{D}_\mathrm{f}\mathbf{Q}\|_F^2, \tag{5.2}$$

ASAP energy

The transformation from the template to the target can involve large deformation. To prevent shear distortion (Figure 5.2), we constrain the deformation matrix $\mathbf{X}_i$ to an orthogonal rotation matrix $\mathbf{R}_i$. As we deal with non-isometric registration, the size between the template and the target can be different. Therefore, a scalar s_i is added to make the deformation scalable [Yamazaki et al. 2013]. The ASAP energy is defined as

$$\begin{aligned}\mathrm{E_{ASAP}}(\mathbf{X}) &= \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{X}_i - s_i\mathbf{R}_i\|_F^2,\\ \text{s.t. } &\mathbf{R}_i^\top\mathbf{R}_i = \mathbf{I}_3, \det(\mathbf{R}_i) > 0,\end{aligned}$$

where $\|\cdot\|_F$ denotes the Forbenius norm, $\mathbf{I}_3$ is a 3×3 identity matrix and $\det(\cdot)$ is the determinant of a matrix. To write the energy in matrix form, we concatenate $s_i, \mathbf{R}_i$ into a $n\times 1$ vector $\mathbf{s} = [s_1\cdots s_n]^\top$ and a $3n\times 3$ matrix $\mathbf{R} \triangleq [\mathbf{R}_1\cdots\mathbf{R}_n]^\top$. The ASAP energy can then be expressed as:

$$\begin{aligned}\mathrm{E_{ASAP}}(\mathbf{X}) &= \frac{1}{2}\|\mathbf{X} - (\mathrm{diag}(\mathbf{s})\otimes\mathbf{I}_3)\mathbf{R}\|_F^2,\\ \text{s.t. } &\mathbf{R}_i^\top\mathbf{R}_i = \mathbf{I}_3, \det(\mathbf{R}_i) > 0,\end{aligned} \tag{5.3}$$

where $\mathrm{diag}(\cdot)$ returns the block-wise diagonal matrix of each row vector in the input matrix, and $\otimes$ denotes the operator of Kronecker product.

Regularization

We assign an affine transformation to each vertex. These transformations are not independent with each other, and the nearby transformations should have overlapping influence. Therefore, the computed transformations should be consistent with respect to one another. In practice, articulated animals only have large transformation deviation at joints,

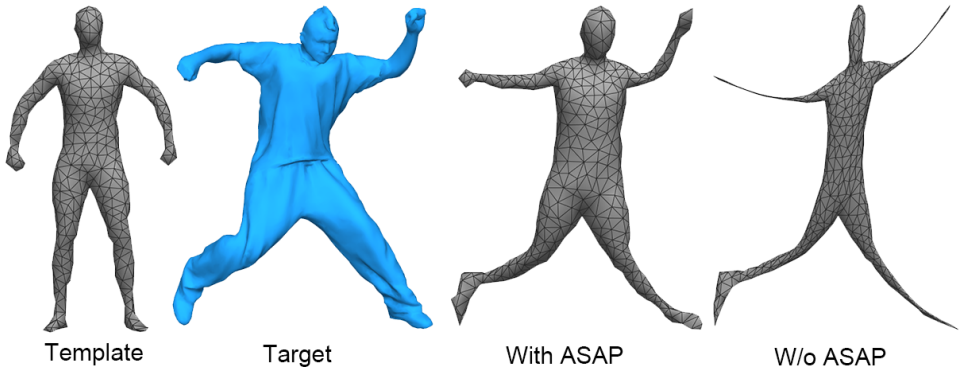


Figure 5.2: *Coarse fitting results with and without ASAP energy. Without ASAP energy, the model can easily get shear distortion.*

so the transformation on the parts between joints should be piecewise smooth with only small deviations. To produce a piecewise smooth result (Figure 5.3), we apply a Huber-norm regularized model constrained on the transformation variation. This metric behaves like an L_2-norm below a certain threshold ε and like an L_1-norm above. The regularization energy is defined as:

$$\mathrm{E_r}(\mathbf{X}, \mathbf{T}) = \sum_{\vec{\mathbf{e}}_{ij} \in \vec{\mathcal{E}}} \|\mathbf{X}_i(\mathbf{p}_j - \mathbf{p}_i) + \mathbf{p}_i + \mathbf{t}_i - (\mathbf{p}_j + \mathbf{t}_j)\|_\varepsilon,$$

where $\vec{\mathbf{e}}_{ij}$ is a directional edge from vertex i to vertex j, $\vec{\mathcal{E}}$ is the half-edge set of the template mesh. Note that an edge has two opposite directional half-edges. $\|\cdot\|_\varepsilon$ is the Huber-norm defined as:

$$\|x\|_\varepsilon = \begin{cases} \frac{\|x\|_2^2}{2\varepsilon}, & \text{if } \|x\|_1 \leq \varepsilon \\ \|x\|_1 - \frac{\varepsilon}{2}, & \text{otherwise} \end{cases},$$

$\varepsilon > 0$ is a threshold, $\|\cdot\|_1, \|\cdot\|_2$ are the L_1, L_2 norm respectively. In order to express the regularization energy with respective to $\mathbf{X}$ and $\mathbf{T}$, we introduce a selection matrix $\mathbf{J} \in \{1\}^{|\vec{\mathcal{E}}| \times n}$ and a directional differential matrix $\vec{\mathbf{K}} \in \{-1, 1\}^{|\vec{\mathcal{E}}| \times n}$. Specifically, each row of $\mathbf{J}$ and $\vec{\mathbf{K}}$ corresponds to a half-edge in $\vec{\mathcal{E}}$ and each column of them corresponds to a vertex in $\mathcal{P}$. Without loss of generality, we assume the r-th rows in $\mathbf{J}$ and $\vec{\mathbf{K}}$ are associated with the half-edge $\vec{\mathbf{e}}_{ij}$. Each row in $\mathbf{J}$ denoted by $\mathbf{J}(r, \cdot)$ only has one non-zero entry, whose column corresponds to the start vertex $\mathbf{p}_i$

of the half edge $\vec{\mathbf{e}}_{ij}$, i.e. $\mathbf{J}(r,i) = 1$. Each row in $\vec{\mathbf{K}}$ denoted by $\vec{\mathbf{K}}(r,\cdot)$ contains two non-zero entries. The entry linked to the reference vertex $\mathbf{p}_i$ is set at -1, while the one linked to the neighbouring vertex $\mathbf{p}_j$ is set at 1, i.e. $\vec{\mathbf{K}}(r,i) = -1$, $\vec{\mathbf{K}}(r,j) = 1$. Then the regularization term can be rewritten as:

$$\mathrm{E_r}(\mathbf{X},\mathbf{T}) = \|\mathrm{diag}(\vec{\mathbf{K}}\mathbf{P})(\mathbf{J}\otimes\mathbf{I}_3)\mathbf{X} - \vec{\mathbf{K}}\mathbf{P} - \vec{\mathbf{K}}\mathbf{T})\|_\varepsilon. \tag{5.4}$$

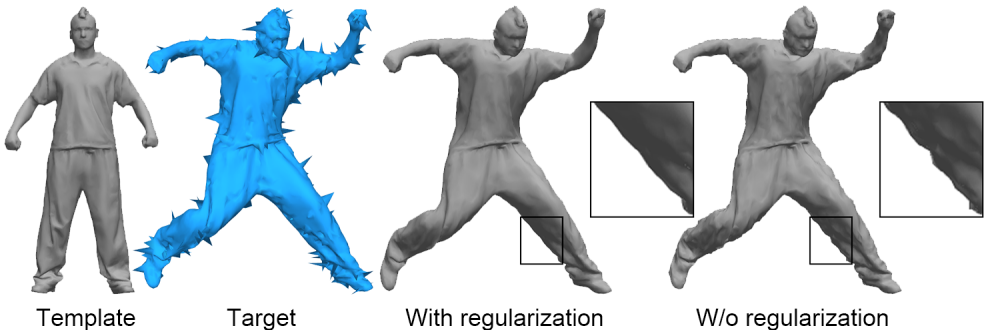


Figure 5.3: *Fitting results with and without regularization energy. The method with regularization energy is more robust against outliers and produces piece-wise smoother result.*

Laplacian energy

To improve the mesh quality, we apply the uniform Laplacian operator on the vertex position, enforcing each vertex to strive to lie in the centroid of its one-ring neighbours and thus the edge lengths strive to be locally equalized (Figure 5.4). The Laplacian energy is defined as follows:

$$\mathrm{E_l}(\mathbf{T}) = \frac{1}{2}\|\mathbf{L}(\mathbf{P}+\mathbf{T})\|_F^2, \tag{5.5}$$

where $\mathbf{L}$ is the uniform Laplacian matrix corresponding to the mesh connectivity:

$$\mathbf{L}(i,j) = \begin{cases} -\frac{1}{d_i}, & \text{if vertices i and j are neighbors,} \\ 1, & \text{if } i = j, \\ 0, & \text{otherwise,} \end{cases}$$

where d_i is the valence of vertex i.

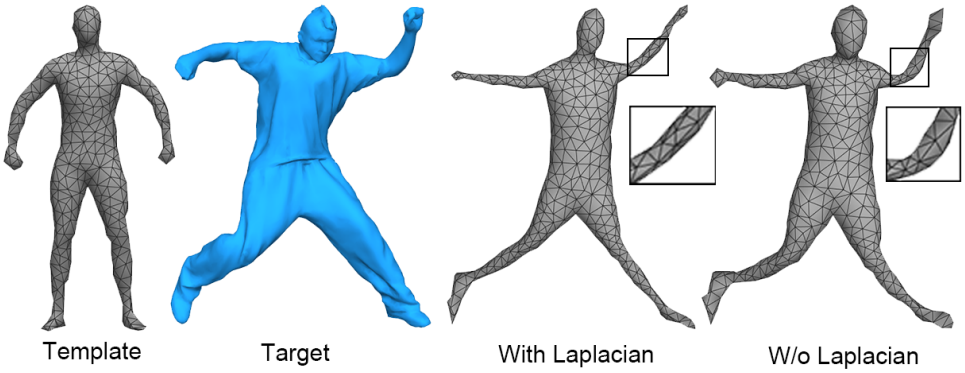


Figure 5.4: *Coarse fitting results with and without Laplacian energy. Compared to the method without Laplacian energy, the variation in the areas of triangle in the method with Laplacian energy is small, specially obvious on the left elbow.*

5.1.3 Mid-scale fitting

This fitting step deforms the coarse mesh as close as possible to the target. Apart from those constraints adopted in the previous step, the data constraint is also applied to attract the coarse mesh towards the target gradually (Figure 5.1f). The energy in this step is denoted by:

$$\mathrm{E}_{\mathrm{mid}}(\mathbf{X}, \mathbf{T}) = \mathrm{E}_{\mathrm{coarse}}(\mathbf{X}, \mathbf{T}) + w_{\mathrm{d}}\mathrm{E}_{\mathrm{d}}. \tag{5.6}$$

Data constraint

To pull the template towards the target, we need to determine the reliable correspondences between the template and the target.
For each vertex $\mathbf{p}_i$ on the template, we project it onto the target along its normal direction. The projection $\mathbf{c}_i$ is regarded as a reliable correspondence only if:

- $\mathbf{c}_i$ is inside a triangle of the target.
- The distance between $\mathbf{c}_i$ and $\mathbf{p}_i$ is under a threshold α.
- The angle between the normals at $\mathbf{c}_i$ and $\mathbf{p}_i$ is under a threshold Θ.

We apply L_1 norm on the position difference to allow a small fraction of regions with large positional error, which is more robust against outliers and fits to the target's details better. The data term is defined as:

$$\mathrm{E_d}(\mathbf{T}) = \sum_{i \in \mathcal{C}} \|\mathbf{p}_i + \mathbf{t}_i - \mathbf{c}_i\|_1,$$

where $\mathcal{C}$ is the template index set of the correspondence. In order to express the data term in regard to $\mathbf{T}$, we stack all $\mathbf{c}_i$ into a matrix $\mathbf{C}$. Similar to $\mathbf{C}_\mathrm{f}$, we define a sparse matrix $\mathbf{C}_\mathrm{d}$ to indicate the corresponding vertices on the template. Therefore, we can rewrite the data term as:

$$\mathrm{E_d}(\mathbf{T}) = \|\mathbf{C}_\mathrm{d}(\mathbf{P} + \mathbf{T}) - \mathbf{C})\|_1. \tag{5.7}$$

5.1.4 Fine fitting

In this step, the dense mesh is first reconstructed from the mid-scale fitting result via embedded deformation method [Sumner et al. 2007] (Figure 5.1g). To avoid foldovers and improve the mesh quality, inspired by [Yeh et al. 2011], the dual-domain relaxation strategy is applied. We alternatively perform the relaxation algorithm between the primal domain and the dual domain. The reconstructed primal mesh is first transformed into its corresponding dual mesh, then the relaxation algorithm is applied on the dual domain (Figure 5.1h). The energy in the dual domain is defined as:

$$\mathrm{E_{dual}}(\mathbf{T}^*) = w_\mathrm{r}\mathrm{E_r}(\mathbf{T}^*) + w_\mathrm{d}\mathrm{E_d}(\mathbf{T}^*) + w_\mathrm{l}\mathrm{E_l}(\mathbf{T}^*). \tag{5.8}$$

After that, the relaxation algorithm is applied in the primary domain (Figure 5.1i), whose energy can be expressed as:

$$\mathrm{E_{primal}}(\mathbf{T}) = w_\mathrm{c}\mathrm{E_c}(\mathbf{T}) + w_\mathrm{l}\mathrm{E_l}(\mathbf{T}), \tag{5.9}$$

where $\mathrm{E_c}$ is the consistent energy transforming the template from the dual domain to the primary domain.

Consistency constraint

It is well-known that one dual mesh corresponds to a primal mesh and their vertices should be consistent: the i-th dual vertex should be equal to the centroid vertex of the i-th triangle of the primal mesh, and two dual vertices corresponding to the adjacent triangles are connected by an edge. (Figure 5.5). The consistency energy is defined as:

$$\mathrm{E_c}(\mathbf{T}) = \frac{1}{2}\sum_{i=1}^{n^*} \|\frac{1}{3}(\mathbf{p}_{i_1} + \mathbf{t}_{i_1} + \mathbf{p}_{i_2} + \mathbf{t}_{i_2} + \mathbf{p}_{i_3} + \mathbf{t}_{i_3}) - \mathbf{p}_i^*\|^2,$$

where $\{i_1, i_2, i_3\}$ are the indices of the primal verices participating in the i-th triangle of the primal mesh. Again, in matrix form, we could rewrite this energy as:

$$\mathrm{E_c}(\mathbf{T}) = \frac{1}{2}\|\mathbf{C_c}(\mathbf{P} + \mathbf{T}) - \mathbf{P}^*\|_F^2, \tag{5.10}$$

where $\mathbf{C_c}$ is a $n^* \times n$ matrix:

$$\mathbf{C_c}(i,j) = \begin{cases} \frac{1}{3}, & \text{if } j \in \{i_1, i_2, i_3\}, \\ 0, & \text{otherwise,} \end{cases}$$

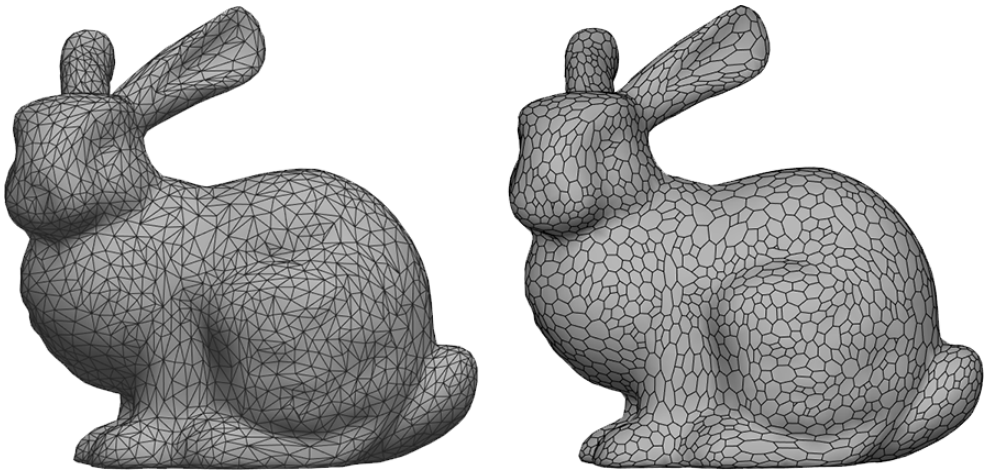


Figure 5.5: *The Stanford bunny's primary mesh and its corresponding dual mesh.*

If the resolution of the template mesh is insufficient to fit the target tightly, a uniform or adaptive subdivision approach can be employed.

Here, we adopt 1-to-4 uniform subdivision method [Lee et al. 2000] to subdivide the template (Figure 5.1j), then the dual-domain relaxation algorithm is performed on the subdivided template mesh again.

5.2 Point cloud registration

The point cloud registration is almost the same as the surface registration except the data constraint. There is no triangle faces in the point cloud of the target, so the template vertices cannot be directly projected on the target to produce their correspondences. Instead, for each vertex $\mathbf{p}_i$ in the template, we first find its closest point $\mathbf{q}_j$ in the target. Similar to Section 4.2, we filter out the pairs if the distance between them exceeds α or if the angle between their normals exceeds a threshold Θ. Then, $\mathbf{q}_j$ is projected on the $\mathbf{p}_i$'s normal vector to yield the correspondence denoted by $Proj(\mathbf{q}_j)$.

In this research, we use Microsoft Kinect v2 to obtain the target point cloud. The Kinect v2 has two cameras (one color camera, one infrared camera) and an infrared illuminator (Figure 5.6). It is based on the time-of-flight (ToF) principle and can offer 512×424 depth images and 1920×1080 color images. The color image and the depth image captured by Kinect v2 can be found in Figure 5.7. Each pixel $\mathbf{u} = (x, y)$ in the depth image $\mathbf{D}$ can be reprojected as a 3D vertex in the camera's coordinated space:

$$\mathbf{v}(\mathbf{u}) = \mathbf{K}^{-1}[x, y, \mathbf{D}(\mathbf{u})]^T. \tag{5.11}$$

Corresponding normal vectors for each vertex are computed by using neighboring reprojected points:

$$\mathbf{n}(\mathbf{u}) = \text{normalize}((\mathbf{v}(x+1, y) - \mathbf{v}(x, y)) \times (\mathbf{v}(x, y+1) - \mathbf{v}(x, y))). \tag{5.12}$$

Then the corresponding color $\mathbf{C}(\mathbf{u})$ in the color image $\mathbf{C}$ will be assigned to each vertex $\mathbf{v}(\mathbf{u})$ to generate the final colored point cloud (Figure 5.8).

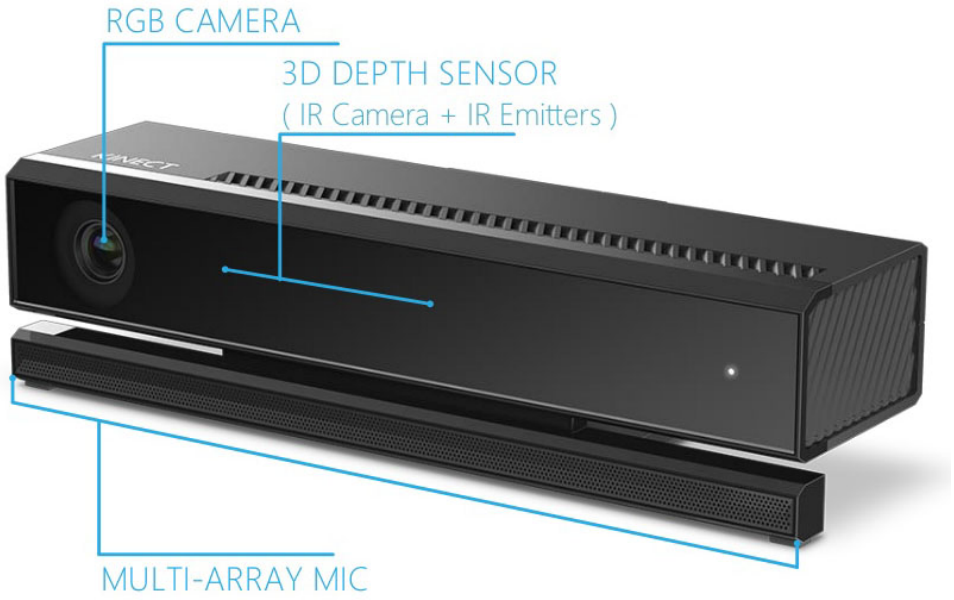


Figure 5.6: *Kinect v2 and its components, including a color camera, an infrared camera and an infrared illuminator.*

(a) **(b)**

Figure 5.7: *(a) A* 512×424 *color image and (b) a* 512×424 *depth image captured by the Kinect v2.*

The feature points can be either specified by users manually or detected by the deep learning technique. We resort to the OpenPose [Cao et al. 2018] to detect 2D COCO pose format feature points in the color image. The COCO pose format consists of 18 feature points of the human body including nose, eyes and other joints, whose location can be seen in Figure 5.9. As the detected feature points by Openpose is in 2D, we need to reproject them back into 3D for 3D shape registration. With the 2D feature points coordinates, the 3D feature points can be com-

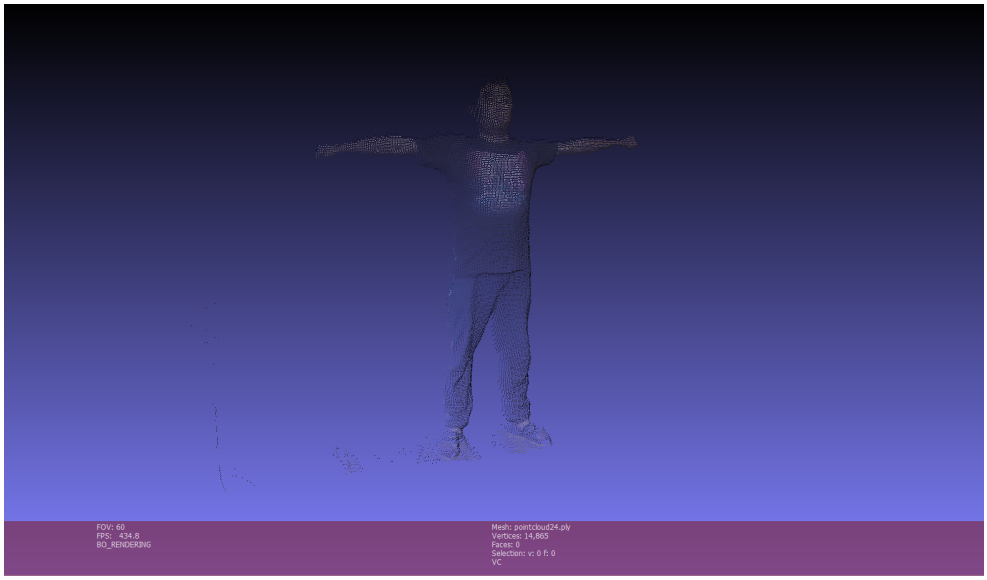


Figure 5.8: *The colored point cloud shown in MeshLab, which is generated from the color image and the depth image in Figure 5.7. Note that the depth truncation is applied on the depth image to extract the person only.*

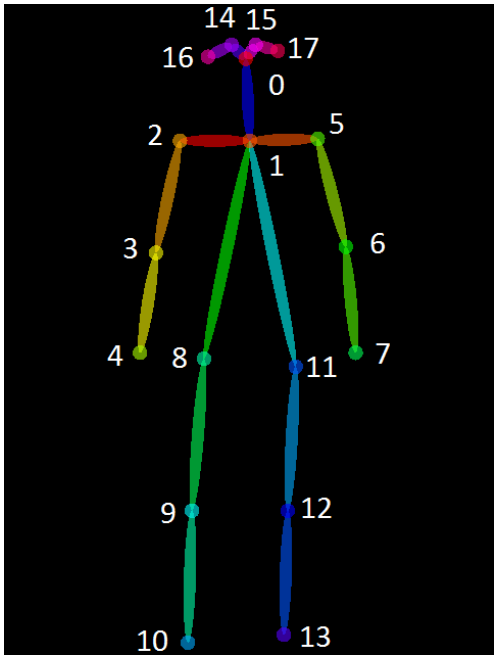
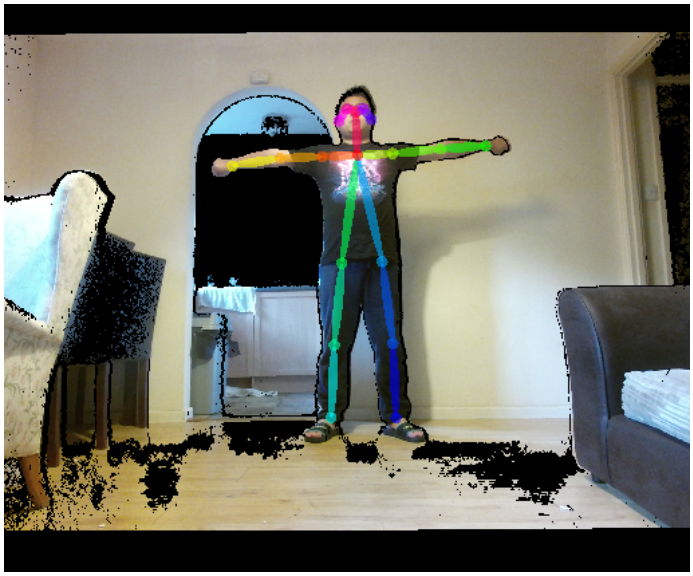


Figure 5.9: *The location of 18 feature points in the COCO pose format.*

puted by equation (5.11) (Figure 5.10). On the template model, we can specify the 3D feature points in advance according to the COCO pose format. In this way, the feature point correspondences can be established automatically in the future (Figure 5.11).



(a)

(b)

Figure 5.10: *(a) 2D feature points detected by OpenPose (b) 3D feature points reprojected from the 2D feature point coordinates.*

With the data constraint and the feature point constraint, the algorithm of surface registration can be easily transplanted to the point cloud registration. The registration result can be found in Figure 5.12.

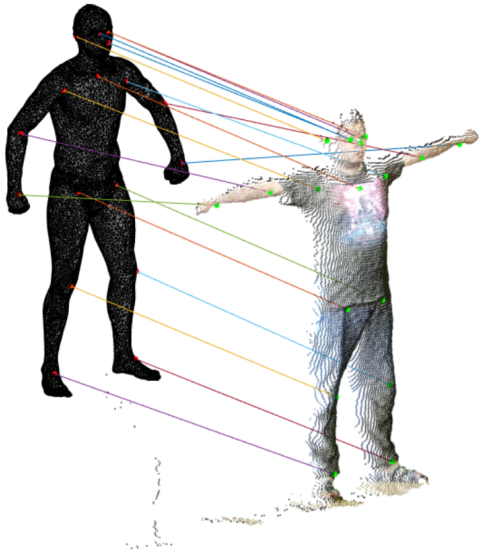
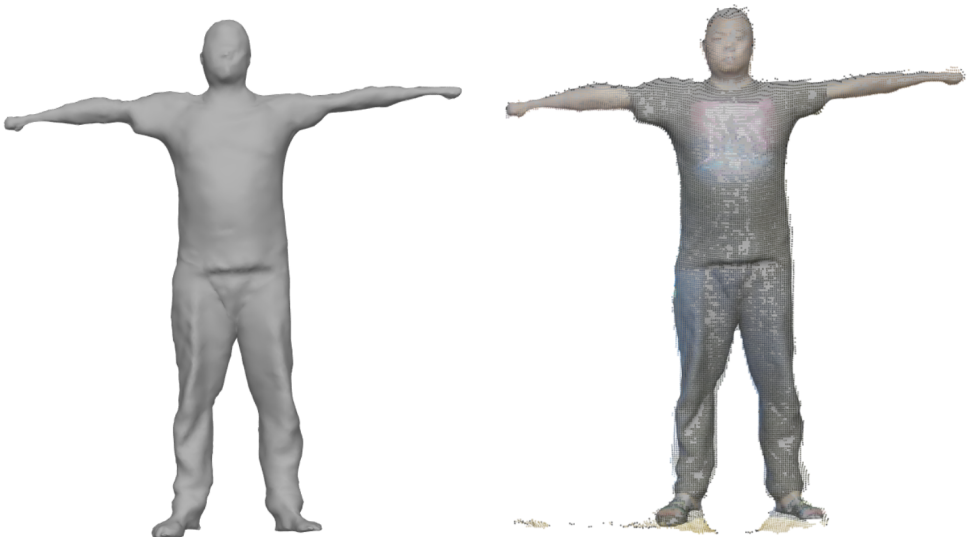


Figure 5.11: *The feature point correspondences between the template and the target point cloud.*



(a) **(b)**

Figure 5.12: *(a) the point cloud registration result (b) the registration result and the target point cloud.*

5.3 Optimization

The optimization process for each fitting step is very similar. The mid-scale step is taken as an example to explain the optimization process as it includes all kinds of norm appeared in the algorithm. Expanding each

term in (5.6) gives us:

$$
\begin{aligned}
\mathrm{E}_{\mathrm{mid}}(\mathbf{X},\mathbf{T}) = & \frac{w_{\mathrm{f}}}{2}\|\mathbf{C}_{\mathrm{f}}(\mathbf{P}+\mathbf{T})-\mathbf{D}_{\mathrm{f}}\mathbf{Q}\|_F^2 \\
& + \frac{w_{\mathrm{ASAP}}}{2}\sum_{i=1}^{n}\|\mathbf{X}_i - s_i\mathbf{R}_i\|_F^2 \\
& + w_{\mathrm{r}}\|\mathrm{diag}(\vec{\mathbf{K}}\mathbf{P})(\mathbf{J}\otimes\mathbf{I}_3)\mathbf{X} - \vec{\mathbf{K}}\mathbf{P} - \vec{\mathbf{K}}\mathbf{T}\|_{\varepsilon} \\
& + \frac{w_{\mathrm{l}}}{2}\|\mathbf{L}(\mathbf{P}+\mathbf{T})\|_F^2 + w_{\mathrm{d}}\|\mathbf{C}_{\mathrm{d}}(\mathbf{P}+\mathbf{T}) - \mathbf{C}\|_1, \\
& \text{s.t. } \mathbf{R}_i^{\top}\mathbf{R}_i = \mathbf{I}_3, \det(\mathbf{R}_i) > 0,
\end{aligned}
\tag{5.13}
$$

To solve the non-differential Huber-norm and L_1-norm, we introduce two auxiliary variables $\mathbf{F}$, $\mathbf{G}$ to change the minimization of (5.13) into the following form:

$$
\begin{aligned}
\min_{\mathbf{X},\mathbf{T},\mathbf{F},\mathbf{G}} & \frac{w_{\mathrm{f}}}{2}\|\mathbf{C}_{\mathrm{f}}(\mathbf{P}+\mathbf{T}) - \mathbf{D}_{\mathrm{f}}\mathbf{Q}\|_F^2 + \frac{w_{\mathrm{ASAP}}}{2}\sum_{i=1}^{n}\|\mathbf{X}_i - s_i\mathbf{R}_i\|_F^2 \\
& + w_{\mathrm{r}}\|\mathbf{F}\|_{\varepsilon} + \frac{w_{\mathrm{l}}}{2}\|\mathbf{L}(\mathbf{P}+\mathbf{T})\|_F^2 + w_{\mathrm{d}}\|\mathbf{G}\|_1, \\
& \text{s.t. } \mathbf{R}_i^{\top}\mathbf{R}_i = \mathbf{I}_3, \det(\mathbf{R}_i) > 0, \\
& \mathbf{F} = \mathbf{H}\mathbf{X} - \vec{\mathbf{K}}\mathbf{P} - \vec{\mathbf{K}}\mathbf{T}, \mathbf{G} = \mathbf{C}_{\mathrm{d}}(\mathbf{P}+\mathbf{T}) - \mathbf{C},
\end{aligned}
\tag{5.14}
$$

where $\mathbf{H} = \mathrm{diag}(\vec{\mathbf{K}}\mathbf{P})(\mathbf{J}\otimes\mathbf{I}_3)$ is introduced for conciseness.

To solve the constrained minimization (5.14), we transform the original problem to iterative minimization of its augmented Lagrangian form:

$$
\begin{aligned}
& L(\mathbf{X},\mathbf{T},\mathbf{F},\mathbf{G},\{\mathbf{R}_i\},\{s_i\},\mathbf{Y},\rho) = \frac{w_{\mathrm{f}}}{2}\|\mathbf{C}_{\mathrm{f}}(\mathbf{P}+\mathbf{T}) - \mathbf{D}_{\mathrm{f}}\mathbf{q}\|_F^2 \\
& + \frac{w_{\mathrm{ASAP}}}{2}\sum_{i=1}^{n}\|\mathbf{X}_i - s_i\mathbf{R}_i\|_F^2 + w_{\mathrm{r}}(\|\mathbf{F}\|_{\varepsilon} + \langle\mathbf{Y}_1, \mathbf{H}\mathbf{X} - \vec{\mathbf{K}}\mathbf{P} - \vec{\mathbf{K}}\mathbf{T} - \mathbf{F}\rangle_F \\
& + \frac{\rho_1}{2}\|\mathbf{H}\mathbf{X} - \vec{\mathbf{K}}\mathbf{P} - \vec{\mathbf{K}}\mathbf{T} - \mathbf{F}\|_F^2) + \frac{w_{\mathrm{l}}}{2}\|\mathbf{L}(\mathbf{P}+\mathbf{T})\|_F^2 + w_{\mathrm{d}}(\|\mathbf{G}\|_1 \\
& + \langle\mathbf{Y}_2, \mathbf{C}_{\mathrm{d}}(\mathbf{P}+\mathbf{T}) - \mathbf{C} - \mathbf{G}\rangle_F + \frac{\rho_2}{2}\|\mathbf{C}_{\mathrm{d}}(\mathbf{P}+\mathbf{T}) - \mathbf{C} - \mathbf{G}\|_F^2), \\
& \text{s.t. } \mathbf{R}_i^{\top}\mathbf{R}_i = \mathbf{I}_3, \det(\mathbf{R}_i) > 0,
\end{aligned}
\tag{5.15}
$$

where ρ_1 and ρ_2 are positve constants, $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are Lagrangian multipliers and $\langle\cdot,\cdot\rangle_F$ is the Forbenius inner product. We solve this problem

by using the ADMM algorithm. The k-th iteration can be summarized as follows:

$$
\begin{cases}
\mathbf{F}^{(k)} = \arg\min_{\mathbf{F}} w_{\mathrm{r}}(\|\mathbf{F}\|_{\varepsilon} + \langle \mathbf{Y}_1^{(k-1)}, \mathbf{H}\mathbf{X}^{(k-1)} - \vec{\mathbf{K}}\mathbf{P} - \vec{\mathbf{K}}\mathbf{T}^{(k-1)} \\
\qquad - \mathbf{F} \rangle_F + \frac{\rho_1}{2}\|\mathbf{H}\mathbf{X}^{(k-1)} - \vec{\mathbf{K}}\mathbf{P} - \vec{\mathbf{K}}\mathbf{T}^{(k-1)} - \mathbf{F}\|_F^2), \\
\mathbf{G}^{(k)} = \arg\min_{\mathbf{G}} w_{\mathrm{d}}(\|\mathbf{G}\|_1 + \langle \mathbf{Y}_2^{(k-1)}, \mathbf{C}_{\mathrm{d}}(\mathbf{P} + \mathbf{T}^{(k-1)}) \\
\qquad - \mathbf{C} - \mathbf{G} \rangle_F + \frac{\rho_2}{2}\|\mathbf{C}_{\mathrm{d}}(\mathbf{P} + \mathbf{T}^{(k-1)}) - \mathbf{C} - \mathbf{G}\|_F^2), \\
\mathbf{R}_i^{(k)} = \arg\min_{\mathbf{R}_i} \frac{w_{\mathrm{ASAP}}}{2}\|\mathbf{X}_i^{(k-1)} - s_i^{(k-1)}\mathbf{R}_i\|_F^2 \\
\qquad \text{s.t. } \mathbf{R}_i^{\top}\mathbf{R}_i = \mathbf{I}_3, \det(\mathbf{R}_i) > 0, \\
s_i^{(k)} = \arg\min_{s_i} \frac{w_{\mathrm{ASAP}}}{2}\|\mathbf{X}_i^{(k-1)} - s_i\mathbf{R}_i^{(k-1)}\|_F^2, \\
\mathbf{X}^{(k)} = \arg\min_{\mathbf{X}} \frac{w_{\mathrm{ASAP}}}{2}\|\mathbf{X} - (\mathrm{diag}(\mathbf{s}^{(k-1)}) \otimes \mathbf{I}_3)\mathbf{R}^{(k-1)}\|_F^2 \\
\qquad + w_{\mathrm{r}}(\langle \mathbf{Y}_1^{(k-1)}, \mathbf{H}\mathbf{X} - \vec{\mathbf{K}}\mathbf{P} - \vec{\mathbf{K}}\mathbf{T}^{(k-1)} - \mathbf{F}^{(k)} \rangle_F \\
\qquad + \frac{\rho_1}{2}\|\mathbf{H}\mathbf{X} - \vec{\mathbf{K}}\mathbf{P} - \vec{\mathbf{K}}\mathbf{T}^{(k-1)} - \mathbf{F}^{(k)}\|_F^2), \\
\mathbf{T}^{(k)} = \arg\min_{\mathbf{T}} \frac{w_{\mathrm{f}}}{2}\|\mathbf{C}_{\mathrm{f}}(\mathbf{P} + \mathbf{T}) - \mathbf{D}_{\mathrm{f}}\mathbf{Q}\|_F^2 \\
\qquad + w_{\mathrm{r}}(\langle \mathbf{Y}_1^{(k-1)}, \mathbf{H}\mathbf{X}^{(k)} - \vec{\mathbf{K}}\mathbf{P} - \vec{\mathbf{K}}\mathbf{T} - \mathbf{F}^{(k)} \rangle_F \\
\qquad + \frac{\rho_1}{2}\|\mathbf{H}\mathbf{X}^{(k)} - \vec{\mathbf{K}}\mathbf{P} - \vec{\mathbf{K}}\mathbf{T} - \mathbf{F}^{(k)}\|_F^2) + \frac{w_{\mathrm{l}}}{2}\|\mathbf{L}(\mathbf{P} + \mathbf{T})\|_F^2 \\
\qquad + w_{\mathrm{d}}(\langle \mathbf{Y}_2^{(k-1)}, \mathbf{C}_{\mathrm{d}}(\mathbf{P} + \mathbf{T}) - \mathbf{C} - \mathbf{G}^{(k)} \rangle_F \\
\qquad + \frac{\rho_2}{2}\|\mathbf{C}_{\mathrm{d}}(\mathbf{P} + \mathbf{T}) - \mathbf{C} - \mathbf{G}^{(k)}\|_F^2) \\
\mathbf{Y}_1^{(k)} = \mathbf{Y}_1^{(k-1)} + \rho_1(\mathbf{H}\mathbf{X}^{(k)} - \vec{\mathbf{K}}\mathbf{P} - \vec{\mathbf{K}}\mathbf{T}^{(k)} - \mathbf{F}^{(k)}), \\
\mathbf{Y}_2^{(k)} = \mathbf{Y}_2^{(k-1)} + \rho_2(\mathbf{C}_{\mathrm{d}}(\mathbf{P} + \mathbf{T}^{(k)}) - \mathbf{C} - \mathbf{G}^{(k)}).
\end{cases}
\tag{5.16}
$$

The $\mathbf{F}$-subproblem has the following closed solution:

$$
\begin{aligned}
\mathbf{F}^{(k)} = & \frac{\varepsilon\rho_1}{\varepsilon\rho_1 + 1}(\mathbf{H}\mathbf{X}^{(k-1)} - \vec{\mathbf{K}}\mathbf{P} - \vec{\mathbf{K}}\mathbf{T}^{(k-1)} + \frac{\mathbf{Y}_1^{(k-1)}}{\rho_1}) \\
& + \frac{1}{\varepsilon\rho_1 + 1} S_{\frac{\varepsilon\rho_1 + 1}{\rho_1}}(\mathbf{H}\mathbf{X}^{(k-1)} - \vec{\mathbf{K}}\mathbf{P} - \vec{\mathbf{K}}\mathbf{T}^{(k-1)} + \frac{\mathbf{Y}_1^{(k-1)}}{\rho_1}),
\end{aligned}
\tag{5.17}
$$

where S is the soft thresholding operator acting on each element of the

given matrix:

$$S_\kappa(a) = \begin{cases} a-\kappa, & \text{if } a > \kappa \\ 0, & \text{if } |a| \leq \kappa \\ a+\kappa, & \text{if } a < -\kappa \end{cases}.$$

The **G**-subproblem can be solved as:

$$\mathbf{G}^{(k)} = S_{\frac{1}{\rho_2}}(\mathbf{C}_\mathrm{d}(\mathbf{P}+\mathbf{T}^{(k-1)}) - \mathbf{C} + \frac{\mathbf{Y}_2^{(k-1)}}{\rho_2}), \tag{5.18}$$

Following the work of [Sorkine & Alexa 2007], we solve the $\mathbf{R}_i$-subproblem by using singular value decomposition of $\mathbf{X}_i$:

$$\mathbf{U}_i \Sigma_i \mathbf{V}_i^T = \mathrm{svd}(\mathbf{X}_i), \mathbf{R}_i^{(k)} = \mathbf{V}_i \mathbf{U}_i^T \tag{5.19}$$

If $\det(\mathbf{R}_i) < 0$, we change the sign of the column of $\mathbf{U}_i$ corresponding to the smallest singular value.

Dividing the s_i-subproblem by s_i and setting its derivative to zero yields:

$$s_i^{(k)} = \frac{\langle \mathbf{X}_i^{(k-1)}, \mathbf{X}_i^{(k-1)} \rangle_F}{\langle \mathbf{R}_i^{(k)}, \mathbf{R}_i^{(k)} \rangle_F} = \frac{\langle \mathbf{X}_i^{(k-1)}, \mathbf{X}_i^{(k-1)} \rangle_F}{3} \tag{5.20}$$

In order to solve the **X**-subproblem, we first zero its derivative and then write the equation in its equally stacked form:

$$\mathbf{A}_x \mathbf{X}^{(k)} = \mathbf{B}_x, \tag{5.21}$$

where

$$\mathbf{A}_x = \begin{pmatrix} \mathrm{sqrt}(w_\mathrm{ASAP})\mathbf{I} \\ \mathrm{sqrt}(w_\mathrm{r})\mathbf{H} \end{pmatrix},$$

$$\mathbf{B}_x = \begin{pmatrix} \mathrm{sqrt}(w_\mathrm{ASAP})(\mathrm{diag}(\mathbf{s}^{(k)}) \otimes \mathbf{I}_3)\mathbf{R}^{(k)} \\ \mathrm{sqrt}(w_\mathrm{r})(\mathbf{F}^{(k)} - \frac{\mathbf{Y}^{(k-1)}}{\rho_1} + \vec{\mathbf{K}}\mathbf{P} + \vec{\mathbf{K}}\mathbf{T}^{(k-1)}) \end{pmatrix},$$

$\mathrm{sqrt}(\cdot)$ is the square root function.

Similarly, $\mathbf{T}$ can be obtained by solving the following equation:

$$\mathbf{A}_t \mathbf{T}^{(k)} = \mathbf{B}_t, \tag{5.22}$$

where

$$\mathbf{A}_t = \begin{pmatrix} \mathrm{sqrt}(w_\mathrm{f})\mathbf{C}_\mathrm{f} \\ \mathrm{sqrt}(w_\mathrm{r})\vec{\mathbf{K}} \\ \mathrm{sqrt}(w_\mathrm{d})\mathbf{C}_\mathrm{d} \\ \mathrm{sqrt}(w_\mathrm{l})\mathbf{L} \end{pmatrix},$$

$$\mathbf{B}_t = \begin{pmatrix} \mathrm{sqrt}(w_\mathrm{f})(\mathbf{D}_\mathrm{f}\mathbf{Q} - \mathbf{C}_\mathrm{f}\mathbf{P}) \\ \mathrm{sqrt}(w_\mathrm{r})(\mathbf{H}\mathbf{X}^{(k)} - \vec{\mathbf{K}}\mathbf{P} - \mathbf{F}^{(k)} + \frac{\mathbf{Y}^{(k-1)}}{\rho_1}) \\ \mathrm{sqrt}(w_\mathrm{d})(\mathbf{G}^{(k)} - \frac{\mathbf{Y}_2^{(k-1)}}{\rho_2} + \mathbf{C} - \mathbf{C}_\mathrm{d}\mathbf{P}) \\ \mathrm{sqrt}(w_\mathrm{l})(-\mathbf{L} * \mathbf{P}) \end{pmatrix}.$$

Now we could summerize the optimization in mid-scale fitting step in Algorithm 5.1. The optimization consists of two loops: the outer loop adjusts the weights and searches the correspondences to construct $\mathbf{E}_\mathrm{d}$, while the inner loop optimizes the translation $\mathbf{T}$ for each template vertex. Once the inner loop converges, $\mathbf{T}$ will be used to update the template vertex position, and then a new outer iteration starts again. In this algorithm, l and k represent the indices of the outer and inner iteration, respectively. The optimization problems in coarse fitting and fine fitting steps can be solved in the same way.

5.4 Experiments

We evaluate the performances of the proposed approach by comparing with state-of-the-art algorithms on clean datasets, noisy datasets and real scans, respectively. The number of feature points, vertices and faces of models used in each example is shown in Table 5.1. The only effort required by users is to specify feature points. All the algorithms are implemented in MATLAB, and all the statistics are measured on an

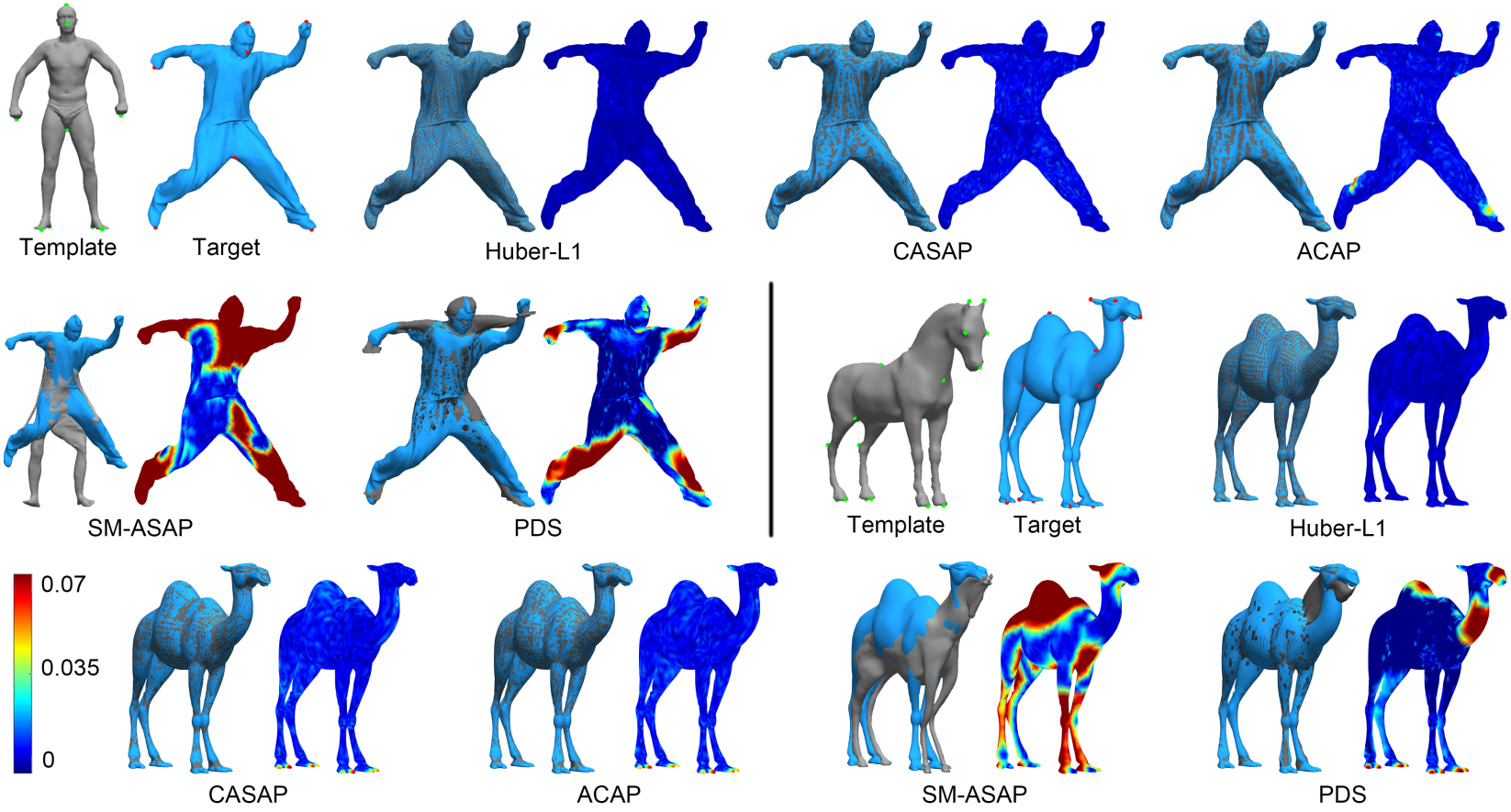


Figure 5.13: *Comparison of non-isometric surface registration approaches on clean data. The color bar denotes the distance from the registration result to the target.*

Algorithm 5.1 Mid-scale fitting step

while not converged **do**
  Adjust weights in (5.13) and construct $\mathbf{E}_{\mathrm{d}}$
  while not converged **do**
    Compute $\mathbf{F}^{(k)}$ by solving equations (5.17).
    Compute $\mathbf{G}^{(k)}$ by solving equations (5.18).
    Compute $\mathbf{R}_i^{(k)}$ by solving equations (5.19).
    Compute $s_i^{(k)}$ by solving equations (5.20).
    Compute $\mathbf{X}^{(k)}$ by solving equations (5.21).
    Compute $\mathbf{T}^{(k)}$ by solving equations (5.22).
    $\mathbf{Y}_1^{(k)}=\mathbf{Y}_1^{(k-1)} + \rho_1(\mathbf{H}\mathbf{X}^{(k)}-\vec{\mathbf{K}}\mathbf{P}-\vec{\mathbf{K}}\mathbf{T}^{(k)}- \mathbf{F}^{(k)})$.
    $\mathbf{Y}_2^{(k)}=\mathbf{Y}_2^{(k-1)}+\rho_2(\mathbf{C}_{\mathrm{d}}(\mathbf{P}+\mathbf{T}^{(k)})-\mathbf{C}-\mathbf{G}^{(k)})$.
  end while
  Update the template by $\mathbf{P}^{(l)} = \mathbf{P}^{(l-1)} + \mathbf{T}^{(l-1)}$.
end while

Intel Xeon E5 3.4 GHz 64-bit workstation with 16GB of RAM.

Name	#FP	Template		Target	
		#V	#F	#V	#F
bouncing	9	12500	24996	10002	20000
camel	24	6608	13200	9469	18934
crane	11	12500	24996	10002	20000
dog	0	25290	50528	25290	50528
gorilla	0	25438	50868	25438	50868
head	10	1669	3298	281581	562554

Table 5.1: *The number of feature points (#FP), vertices (#V), faces (#F) of the template and the target models in the examples.*

5.4.1 Parameters and weights

In the data constraints, we set $\Theta = 90°$, $\alpha = 0.05 r_{\mathrm{box}}$, where r_{box} is the length of the target bounding box diagonal. In the regularization term, $\epsilon = 0.1$ is used, $\rho = 1$ for the optimization. As for the weights, we use $w_{\mathrm{ASAP}} = 1$, $w_{\mathrm{r}} = 10$, $w_{\mathrm{f}} = 1000$, $w_{\mathrm{l}} = 10$ for the whole coarse fitting step to highlight the feature point constraints. In the mid-scale step, w_{d} is initialized to 0.1 then increased by 0.1 for each outer iteration. w_{l} is initialized to 100 then decreased by 10 for each outer iteration until reaching 1. We gradually increase w_{d} and reduce w_{l} so that the good quality of the template mesh can be maintained during the registration to

avoid the foldover occurrence. The other weights are the same as those in the coarse fitting step. In the fine fitting step, we set $w_{\mathrm{d}} = 0.1$, $w_{\mathrm{r}} = 0.1$ for (5.8); $w_{\mathrm{c}} = 1000$ for (5.9); w_{l} is initialized to 100 then decreased by 10 for each outer iteration until reaching 1 through the whole fine fitting step; w_{d} is increased by 0.1 for each outer iteration as the same reason in the mid-scale step. For different models, parameters could be slightly different, but the general idea behind it is the same: in coarse fitting, the feature point constraint is dominant to roughly adjust the template pose and size to match the feature points. While in mid-scale and fine-scale fitting, the weights on data constraint gradually increase to attract the template towards the target from slow to fast.

5.4.2 Results on clean data

Our method is compared with state-of-the-art non-isometric registration methods (Figure 5.13): consistent as-similar-as-possible surface registration (CASAP) [Jiang et al. 2017], as-conformal-as-possible surface registration (ACAP) [Yoshiyasu et al. 2014], the shape matching based registration method that minimizes the as-similar-as-possible energy (SM-ASAP) [Papazov & Burschka 2011] and the registration method that utilizes the point-based deformation smoothness regularization (PDS) [Amberg et al. 2007]. To evaluate the registration accuracy quantitatively, we follow the criterion used in [Jiang et al. 2017; Li et al. 2018] and measure (1) distance error, which is the average distance from the vertices of the deformed template to the corresponding points on the target relative to the target bounding box diagonal, (2) intersection error, which is the number of self-intersecting faces, (3) Hausdorff error, the largest distance between two shapes with respective to the target bounding box diagonal. The statistics data can be found in Table 5.2. It is obvious that our method Huber-L_1 produces least errors without any foldover generated. Although CASAP and ACAP can fit the template close to the target, as shown at the left ankle of the bouncing model, they still suffer from the foldover issue when the deformation is dramatic. SM-ASAP does not require feature points, but it is only able to handle surfaces with close initial alignment and similar poses. PDS

allows for affine transformation for each vertex, which makes it too weak against shear distortions as shown at both arms of the bouncing model. Table 5.3 shows the iteration steps and time taken by each method. The total time cost in SM-ASAP and PDS are relatively small, however, their results are undesirable with large errors and many foldovers. The total time used in the rest three methods are almost on the same level, but our method gets the best result with no foldover occured.

		Huber-L_1	CASAP	ACAP	SM-ASAP	PDS
bouncing	D	**2.1556e-05**	4.3619e-05	4.9460e-05	0.0010	1.8525e-04
	I	**0**	242	494	909	5639
	H	**0.0113**	0.0273	0.0784	0.7279	0.1619
camel	D	**7.1209e-05**	1.8185e-04	1.8074e-04	9.8233e-04	2.9304e-04
	I	**0**	**0**	76	85	3884
	H	**0.0172**	0.0704	0.0710	0.3222	0.1238

Table 5.2: *Quantitative evaluation in the bouncing and camel examples. D, I and H indicate distance error [%], intersection error and Hausdorff error [%] respectively.*

		Huber-L_1	CASAP	ACAP	SM-ASAP	PDS
bouncing	#O	24	54	73	**9**	500
	#I	2064	3477	577	**37**	1608
	Inner	0.068	**0.034**	0.277	0.252	0.035
	Total	140.199	118.044	159.687	**9.346**	56.281
camel	#O	24	54	73	**9**	500
	#I	2037	3352	562	**34**	1582
	Inner	0.065	**0.032**	0.272	0.261	0.034
	Total	132.405	107.264	152.864	**8.874**	54.328

Table 5.3: *Iteration steps and time (in seconds) in the bouncing and camel example. #O, #I indicate the number of outer iteration steps and total inner iteration steps respectively. "Inner" indicates the average time required for each inner iteration step. "Total" represents the total fitting time.*

5.4.3 Results on noisy data

In this subsection, we set up two different experiments to demonstrate the robustness of our method. In both experiments, the targets are polluted with noise along the normal direction of each vertex by multiplying the standard deviation of the average length of the edges in the target.

Firstly, we compare our method with the start-of-the-arts on the noisy

data in Figure 5.14. The quantitative evaluation is shown in Table 5.4. Affected by the noise, CASAP and ACAP get poor initial shape estimation and regard some noise as correspondence, which makes parts of template fitted to noise as shown at the right waist of CASAP and the left arm of ACAP. SM-ASAP and PDS still produce poor results as on the clean data. Thanks to the dual relaxation and Huber-L_1 regularization, our method is robust against noise and achieves more accurate results than other methods without any foldover generated.

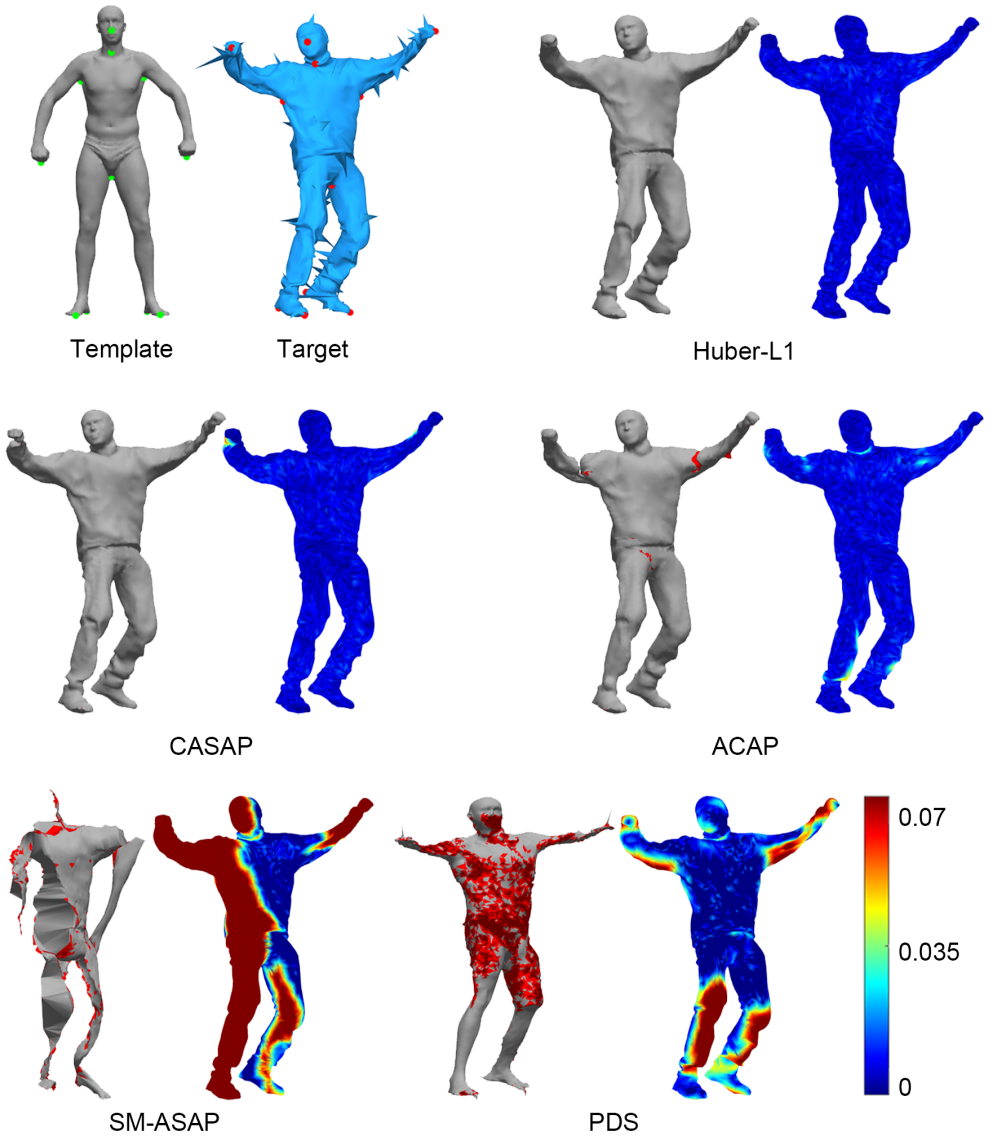


Figure 5.14: *Comparison of non-isometric surface registration approaches on noisy data. The self-intersection faces on the template are colored in red.*

		Huber-L_1	CASAP	ACAP	SM-ASAP	PDS
crane	D	**4.7397e-05**	5.2738e-05	5.6536e-05	9.4177e-04	1.6583e-04
	I	**0**	143	453	8965	5862
	H	**0.0227**	0.0729	0.0469	0.5691	0.1258

Table 5.4: *Quantitative evaluation in the crane example.*

Secondly, instead of comparing with a particular method (e.g. CASAP), we compare our Huber-L_1 with different norms applied on the regularization term and data term, including L_1-L_1, L_1-L_2 and L_2-L_2, as shown in Figure 5.15. The other terms are fixed and the whold pipeline is identical. To ensure fair comparison, the correspondences among different approaches are the same and have been given as priors. As the L_2-norm is easily affected by the outliers, L_1-L_2 and L_2-L_2 produce larger errors than Huber-L_1 and L_1-L_1, especially at the places with large deformation (seen at the shoulder and butt of the gorilla in SNR and L_2-L_2). Due to the L_1-norm's tendency on favoring sparse solution, the effect caused by the regularizer leads to piecewise constant solutions (shown at the tail of the dog in L_1-L_1). This effect can be reduced significantly by using a quadratic penalization for small gradient magnitudes while sticking to linear penalization for larger magnitudes to maintain the discontinuity properties known from total variation, which is the Huber-norm we adopted here. The quantitative evaluation is shown in Table 5.5. As the benchmark models also involve fold-overs, we ignore the intersection error in the table. With Huber-L_1 regularization scheme, Huber-L_1 is not only robust against noise but also produces piecewise smooth results with least errors.

		Huber-L_1	L_1-L_1	L_1-L_2	L_2-L_2
dog	D	**1.0441e-06**	5.2783e-06	1.0682e-05	1.5101e-05
	H	**1.6409**	2.9771	3.9666	2.6419
gorilla	D	**2.3474e-06**	2.8365e-06	1.6977e-05	2.3565e-05
	H	**4.1795**	4.5067	5.5522	5.1793

Table 5.5: *Quantitative evaluation in the dog and gorilla examples.*

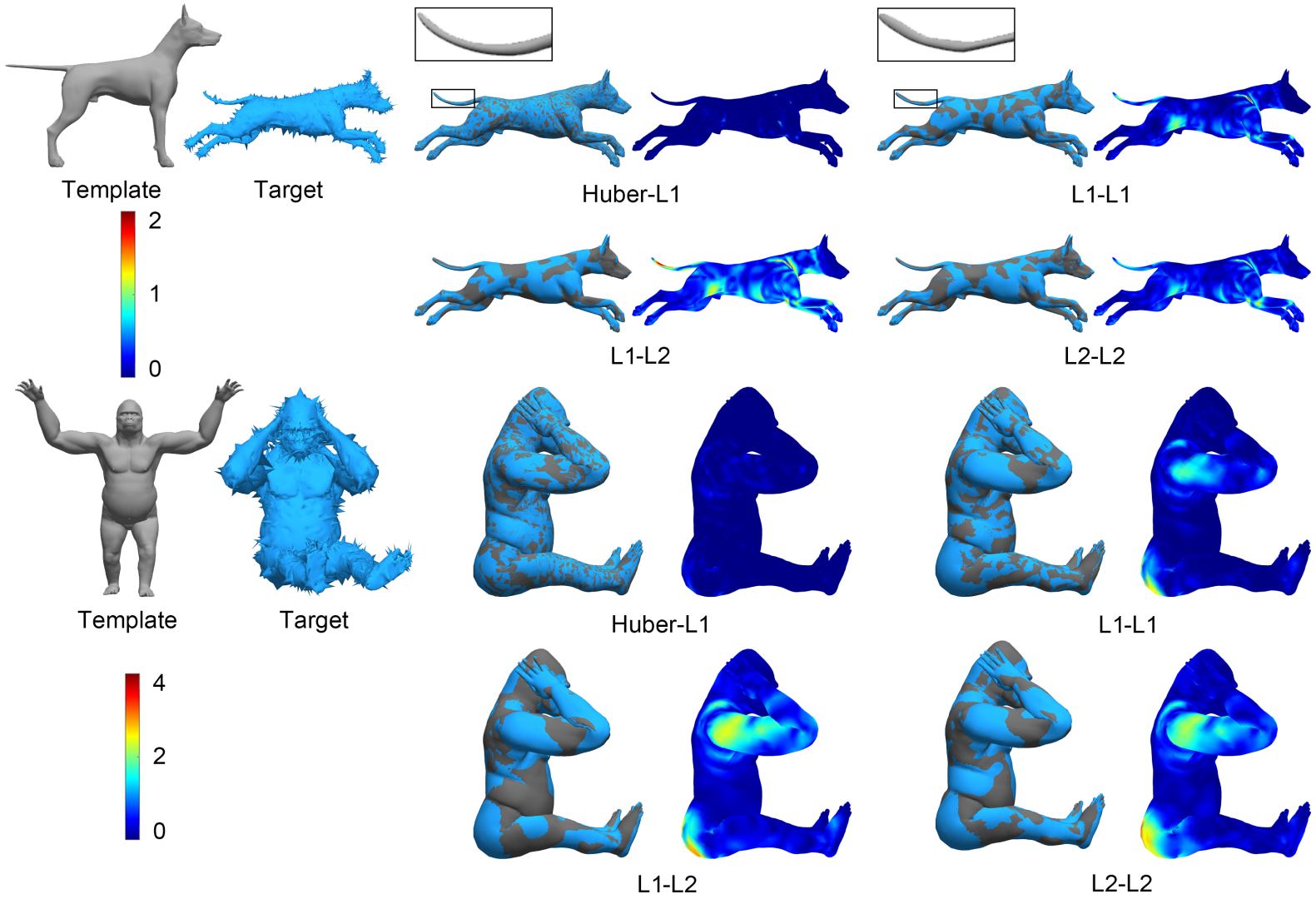


Figure 5.15: *Comparison of different norms applied on the regularization term and data term on noisy data.*

5.4.4 Results on real scans

Finally, the Huber-L_1 method is compared with the state-of-the-art approaches on real scan in Figure 5.16. The quantitative evaluation is shown in Table 5.6. In terms of distance error and Hausdorff error, CASAP and ACAP have competitive results with our method. However, they still produce foldovers at the canthus and nostril, and especially generate poor results around the noisy boundary and the missing parts of the target (seen from the side view). Compared to CASAP and ACAP, SM-ASAP and PDS produce less foldovers, however, they have larger distance errors and Hausdorff errors so that their results even look dissimilar to the target intuitively. On the contrary, Huber-L_1 produces least errors with no foldover generated, which demonstrates our method is more robust to a noisy and incomplete target.

		Huber-L_1	CASAP	ACAP	SM-ASAP	PDS
head	D	**3.6954e-06**	4.0682e-04	3.1503e-04	0.0016	0.0048
	I	**0**	1287	2345	103	185
	H	**3.1624**	3.2413	3.2244	3.2463	3.2516

Table 5.6: *Quantitative evaluation in the head example.*

5.5 Summary

In this chapter, we have proposed a novel non-isometric surface registration approach based on the Huber-L_1 model. The Huber-norm regularizes on transformation variation, which is robust to noise and produces piecewise smooth result. The position difference is regularized in L_1-norm, which preserves the fine details on the target while smoothing the noise. The ASAP energy allows us to handle shapes in different sizes. To improve the efficiency and robustness of registration, a coarse-to-fine strategy is adopted. The Laplacian energy relaxes the template on the primal and dual domain, reducing self-intersection and foldover occurrence and improving the mesh quality. The experiments on various data have shown our method is more robust and accurate than other state-of-the-art approaches.

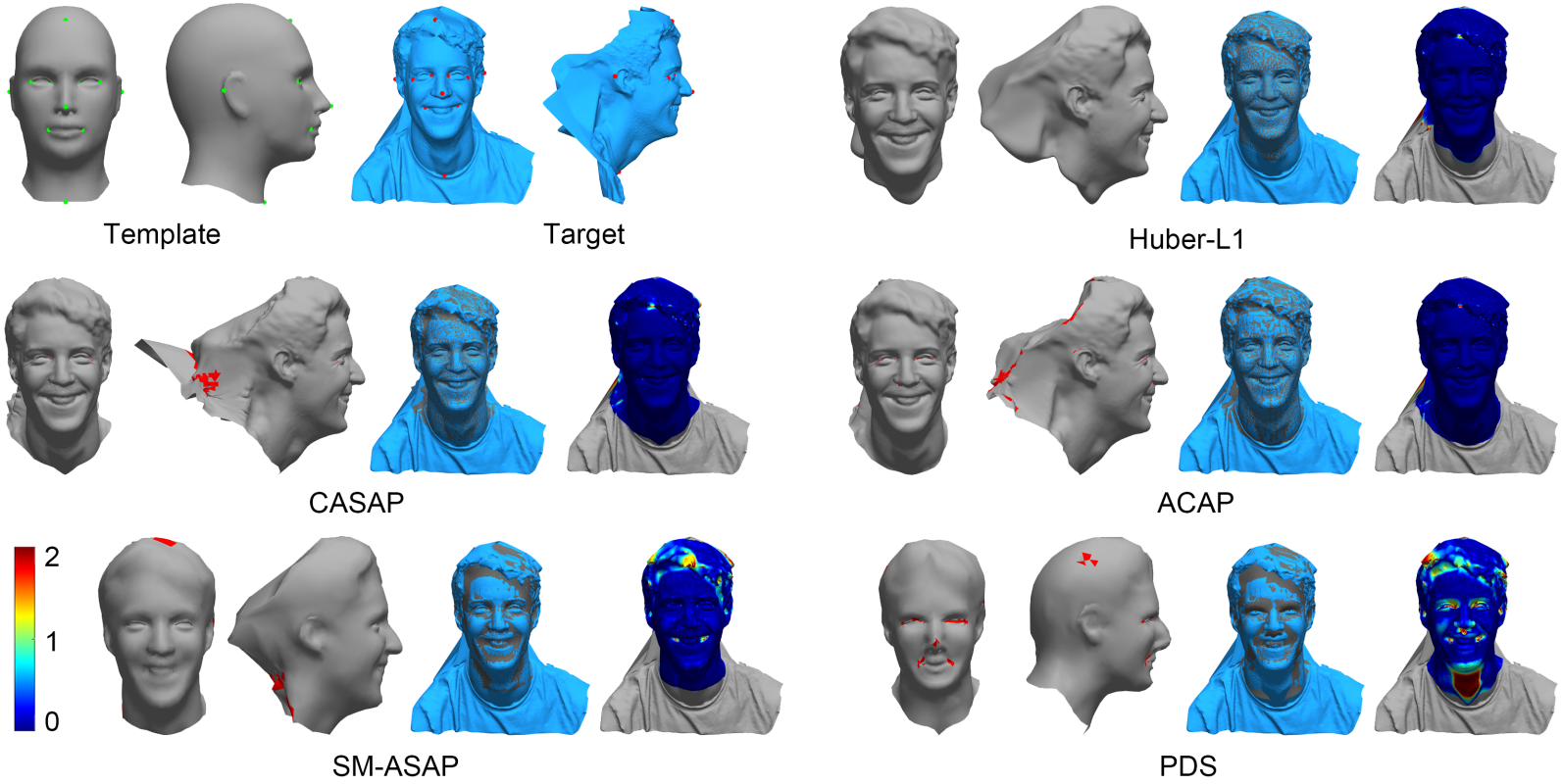


Figure 5.16: *Comparison of non-isometric surface registration approaches on real scan.*

Chapter 6

Conclusion and Future works

6.1 Conclusions

This thesis proposed non-isometric 3D shape registration methods aiming at solving the main challenges in Section 1.2, i.e. template quality preservation, semantic correspondence, registration robustness and less user effort. Chapter 2 gave an overview of existing works in registration related areas, including 3D geometric deformation, 3D shape correspondence, 3D shape registration and registration robustness. It discussed the current research progress and then analyzed the main challenges remained in these works, which provided guideline for our following research.

In Chapter 3, a novel consistent as-similar-as-possible deformation method is proposed. we analyzed the pros and cons of ARAP, SR-ARAP and ASAP deformation methods. The ARAP method is not scalable, nor its energy is consistent. The SR-ARAP method makes up the bending energy achieving a consistent energy, while it is still not able to handle local scalability. The ASAP method allows local scalability but its energy is not consistent. We combined the benefits of SR-ARAP and the advantages of ASAP, came up with a CASAP deformation method, which not only allows local scale to each discrete cell but also achieves the consistent discretization for surfaces without compromising the efficiency. We then compared ASAP and CASAP with the ground truth consistent method

(VASAP), the experiments have shown that our method produced closer results to the ground truth. Finally, we deformed the same shapes but with different discretization and evaluated the distance between them. Compared with ASAP, our consistent method produced less errors and resulted in very similar qualitative behaviors on different discretization.

In Chapter 4, based on the CASAP deformation method proposed in Chapter 3, a novel non-isometric surface registration method (CASAP) has been presented. The registration includes three different constraints: CASAP deformation constraint, correspondence constraint and feature point constraint. The CASAP deformation constraint enables us to deform the template in a consistent as-similar-as-possible way. In Section 4.2, we not only considered the vertices' positions but also their feature descriptors such as multi-scale mean curvatures, WHS, HKS to choose more accurate correspondences than works only choose the closest vertices as correspondences. Compared to the correspondence constraint, the feature points give us more semantically accurate correspondences, which are employed to adjust the overall size and pose of the template. In Subsection 4.3.3, a coarse-to-fine registration scheme has been proposed to further improve the registration efficiency and accuracy. The experiments have shown that CASAP registration method produced more accurate fitting results and required less user efforts compared to the state-of-the-arts.

In chapter 5, we have proposed a novel non-isometric surface registration approach based on the Huber-L_1 model. The Huber-norm regularizes on transformation variation, which is robust to noise and produces piecewise smooth result. The position difference is regularized in L_1-norm, which preserves the fine details on the target while smoothing the noise. The ASAP energy allows us to handle shapes in different sizes. To improve the efficiency and robustness of registration, a coarse-to-fine strategy is adopted. The Laplacian energy relaxes the template on the primal and dual domain, reducing self-intersection and fold-over occurrence and improving the mesh quality. The experiments on various data have shown our method is more robust and accurate than other state-of-the-art approaches.

6.2 Future works

6.2.1 No fold-over guarantee

Although the CASAP and Huber-L_1 registration methods produced more accurate results than the state-of-the-arts, they cannot guarantee to avoid the fold-overs and self-intersections completely, especially for parts with large curvature. It is a common issue in surface registration methods since the inverted elements in 3D surface cannot be measured and thus be detected. To solve this problem, a tetrahedron template model will be used so that the inverted elements can be detected and then prevented in advance to avoid fold-overs locally.

In surface modeling and physics-based animation, a common way to avoid inversion elements is to design penalizing inversion constraints [Irving et al. 2004; Chao et al. 2010; Stomakhin et al. 2012; Setaluri et al. 2014; Civit-Flores & Susín 2014]. There are some specialized material models such as Neo-hookean elasticity [Bonet & Wood 1997] contain energies that increase to infinity as the area/volume of a deformed element degenerates to zero. Unfortunately, the numeric solutions to these energies are extremely complicate and may bring the classical Newton's method to a halt [Schüller et al. 2013]. In the context of inverse elastic shape design Chen et al. [2014] discusses how to deal with these numerical complexities. Schüller et al. [2013] propose Locally Injective Mapping (LIM), they apply a custom barrier function which results in real-time feedback to the user. Jin et al. [2014] improved LIM subsequently by online remeshing. LIM guarantees that no inverse elements will produce as long as the initial input configuration is inversion-free. Poranne & Lipman [2014] present an interactive inversion-free deformation approach with provable guarantees, however, their method is only limited to 2D deformations. Based on projections onto approximate tangent planes, Kovalsky et al. [2015] provide an efficient algorithm to calculate bounded distortion mappings.

Since locally injective mapping guarantees that there is no inversion element as long as the initial input configuration is inversion-free, fol-

lowing the work of [Schüller et al. 2013; Liu et al. 2016], we will add a barrier function into the total energy to penalize the inverted elements so that no fold-over occurrence can be guaranteed, which is a critical property for registration robustness.

6.2.2 Dynamic registration

In this research, the registration instances are static and separated. Only one template and one target (no matter in the form of surface or point cloud) are used in one registration. However, if the target is continuously captured by a scanning system, the information in each frame is limited and the neighboring frames should share some common information of the target.

There are many existing methods which reconstruct a deforming model dynamically. Liao et al. [2009] develop a novel approach to reconstruct complete 3D surface deformation over time by a single camera. The deformable surface patches are stitched together by mesh deformation in a global manner, and merged into a complete model by a volumetric method. Dou et al. [2015]present a system which merges a sequence of images from a single range sensor into a unified 3D model, without requiring an initial template. Xu et al. [2015]introduce a template-less 4D reconstruction method that incrementally fuses highly-incomplete 3D observations of a deforming object, and generates a complete, temporally coherent shape representation of the object. Yu et al. [2018] propose a new real-time system that combines volumetric dynamic reconstruction with data-driven template fitting to simultaneously reconstruct detailed geometry, non-rigid motion and the inner human body shape from a single depth camera.

In the future, based on the existing methods, a novel dynamic registration method will be proposed, which is not only able to register the template in single frame but also fuse every frame's registration results into a global and unified model. Equipped with joints detection technique, it will be easy to extract the dynamic skeleton for the model. Dynamic models with skeleton will bring out a lot of conscience for ani-

mation industry.

6.2.3 Registration with highly detailed facial expression

In Section 5.2, the template has been successfully registered to the point cloud target captured by the Kinect depth camera. However, restricted to the depth map resolution, the detailed facial expression cannot be well presented by the coarse point cloud. Therefore, it is a big challenge to make the registration results look like the original target.

Cao et al. [2015] present the first method capable of capturing facial performances in real-time at high fidelity, including medium scale details such as wrinkles. Richardson et al. [2017] propose an end-to-end approach for detailed face reconstruction from a single image. Bogo et al. [2015] accurately estimate the 3D geometry and appearance of the human body from a monocular RGB-D sequence of a user moving freely in front of the sensor. They develop a new parametric 3D body model, call Delta, which is based on SCAPE dataset [Anguelov et al. 2005a]. Hesse et al. [2019] present a method for learning a statistical 3D Skinned Multi-Infant Linear body model (SMIL) from incomplete, low-quality RGB-D sequences of freely moving infants.

In the future, instead of using the depth map to infer the facial expression, we will resort to deep learning technique to track the facial performance just from the RGB images and then reconstruct the 3D facial model in a form of linear combination of blendshapes. Finally, the high detailed facial model will be fused back into the whole body registration result. This will give people a more realistic experience.

Bibliography

D. Aiger, et al. (2008). '4-points congruent sets for robust pairwise surface registration'. *ACM Transactions on Graphics (TOG)* **27**(3):85.

B. Allen, et al. (2003). 'The space of human body shapes: reconstruction and parameterization from range scans'. *ACM transactions on graphics (TOG)* **22**(3):587–594.

B. Amberg, et al. (2007). 'Optimal step nonrigid ICP algorithms for surface registration'. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8. IEEE.

A. Angelidis, et al. (2006). 'Swirling-sweepers: Constant-volume modeling'. *Graphical Models* **68**(4):324–332.

D. Anguelov, et al. (2005a). 'SCAPE: shape completion and animation of people'. In *ACM Transactions on Graphics (TOG)*, vol. 24, pp. 408–416. ACM.

D. Anguelov, et al. (2005b). 'The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces'. In *Advances in neural information processing systems*, pp. 33–40.

O. K.-C. Au, et al. (2007). 'Handle-aware isolines for scalable shape editing'. In *ACM Transactions on Graphics (TOG)*, vol. 26, p. 83. ACM.

M. Aubry, et al. (2011). 'The wave kernel signature: A quantum mechanical approach to shape analysis'. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 1626–1633. IEEE.

M. A. Audette, et al. (2000). 'An algorithmic overview of surface registration techniques for medical imaging'. *Medical image analysis* **4**(3):201–217.

S. Belongie, et al. (2001). 'Shape context: A new descriptor for shape matching and object recognition'. In *Advances in neural information processing systems*, pp. 831–837.

S. Belongie, et al. (2002). 'Shape matching and object recognition using shape contexts'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(4):509–522.

P. J. Besl & N. D. McKay (1992). 'A method for registration of 3-D shapes'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(2):239–256.

S. Biasotti, et al. (2006). 'Sub-part correspondence by structural descriptors of 3D shapes'. *Computer-Aided Design* **38**(9):1002–1019.

B. Bickel, et al. (2008). 'Pose-space animation and transfer of facial details'. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 57–66. Eurographics Association.

F. Bogo, et al. (2015). 'Detailed full-body reconstructions of moving people from monocular RGB-D sequences'. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2300–2308.

J. Bonet & R. D. Wood (1997). *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press.

M. Botsch & L. Kobbelt (2003). 'Multiresolution surface representation based on displacement volumes'. In *Computer Graphics Forum*, vol. 22, pp. 483–491. Wiley Online Library.

M. Botsch & L. Kobbelt (2004). 'An intuitive framework for real-time freeform modeling'. In *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 630–634. ACM.

M. Botsch & L. Kobbelt (2005). 'Real-Time Shape Editing using Radial

Basis Functions'. In *Computer graphics forum*, vol. 24, pp. 611–621. Wiley Online Library.

M. Botsch, et al. (2006). 'PriMo: coupled prisms for intuitive surface modeling'. In *Symposium on Geometry Processing*, no. EPFL-CONF-149310, pp. 11–20.

M. Botsch, et al. (2007). 'Adaptive space deformations based on rigid cells'. In *Computer Graphics Forum*, vol. 26, pp. 339–347. Wiley Online Library.

M. Botsch & O. Sorkine (2007). 'On linear variational surface deformation methods'. *IEEE transactions on visualization and computer graphics* **14**(1):213–230.

M. Botsch & O. Sorkine (2008). 'On linear variational surface deformation methods'. *IEEE Transactions on Visualization and Computer Graphics* **14**(1):213–230.

S. Bouaziz, et al. (2013). 'Sparse iterative closest point'. In *Computer graphics forum*, vol. 32, pp. 113–123. Wiley Online Library.

S. Boyd, et al. (2011). 'Distributed optimization and statistical learning via the alternating direction method of multipliers'. *Foundations and Trends in Machine learning* **3**(1):1–122.

M. M. Bronstein & I. Kokkinos (2010). 'Scale-invariant heat kernel signatures for non-rigid shape recognition'. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1704–1711. IEEE.

C. Cao, et al. (2015). 'Real-time high-fidelity facial performance capture'. *ACM Transactions on Graphics (ToG)* **34**(4):46.

Z. Cao, et al. (2018). 'OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields'. In *arXiv preprint arXiv:1812.08008*.

U. Castellani, et al. (2008). 'Sparse points matching by combining 3D mesh saliency with statistical descriptors'. In *Computer Graphics Forum*, vol. 27, pp. 643–652. Wiley Online Library.

G. Celniker & D. Gossard (1991). 'Deformable curve and surface finite-elements for free-form shape design'. *ACM SIGGRAPH computer graphics* **25**(4):257–266.

A. Chambolle & T. Pock (2011). 'A first-order primal-dual algorithm for convex problems with applications to imaging'. *Journal of mathematical imaging and vision* **40**(1):120–145.

W. Chang & M. Zwicker (2008). 'Automatic registration for articulated shapes'. In *Computer Graphics Forum*, vol. 27, pp. 1459–1468. Wiley Online Library.

I. Chao, et al. (2010). 'A simple geometric model for elastic deformations'. *ACM Transactions on Graphics (TOG)* **29**(4):38.

X. Chen, et al. (2014). 'An asymptotic numerical method for inverse elastic shape design'. *ACM Transactions on Graphics (TOG)* **33**(4):95.

H. Chui & A. Rangarajan (2003). 'A new point matching algorithm for non-rigid registration'. *Computer Vision and Image Understanding* **89**(2):114–141.

O. Civit-Flores & A. Susín (2014). 'Robust treatment of degenerate elements in interactive corotational fem simulations'. In *Computer Graphics Forum*, vol. 33, pp. 298–309. Wiley Online Library.

S. Coquillart (1990). *Extended free-form deformation: a sculpturing tool for 3D geometric modeling*, vol. 24. ACM.

M. Dou, et al. (2015). '3D scanning deformable objects with a single RGBD sensor'. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 493–501.

A. Elad & R. Kimmel (2003). 'On bending invariant signatures for surfaces'. *IEEE Transactions on pattern analysis and machine intelligence* **25**(10):1285–1295.

M. S. Floater & K. Hormann (2005). 'Surface parameterization: a tutorial and survey'. In *Advances in multiresolution for geometric modelling*, pp. 157–186. Springer.

R. Gal & D. Cohen-Or (2006). 'Salient geometric features for partial shape matching and similarity'. *ACM Transactions on Graphics (TOG)* **25**(1):130–150.

L. Gao, et al. (2016). 'Efficient and flexible deformation representation for data-driven surface modeling'. *ACM Transactions on Graphics (TOG)* **35**(5):1–17.

L. Gao, et al. (2019). 'Sparse data driven mesh deformation'. *IEEE transactions on visualization and computer graphics* .

T. Gatzke, et al. (2005). 'Curvature maps for local shape comparison'. In *International Conference on Shape Modeling and Applications 2005 (SMI'05)*, pp. 244–253. IEEE.

N. Gelfand, et al. (2005). 'Robust global registration'. In *Symposium on geometry processing*, vol. 2, p. 5.

B. Gilles, et al. (2010). 'Creating and Animating Subject-Specific Anatomical Models'. In *Computer Graphics Forum*, vol. 29, pp. 2340–2351. Wiley Online Library.

I. Guskov, et al. (1999). 'Multiresolution signal processing for meshes'. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 325–334. ACM Press/Addison-Wesley Publishing Co.

N. Hesse, et al. (2019). 'Learning and Tracking the 3D Body Shape of Freely Moving Infants from RGB-D sequences'. *IEEE transactions on pattern analysis and machine intelligence* .

K. Hormann & N. Sukumar (2008). 'Maximum entropy coordinates for arbitrary polytopes'. In *Computer Graphics Forum*, vol. 27, pp. 1513–1520. Wiley Online Library.

W. M. Hsu, et al. (1992). 'Direct manipulation of free-form deformations'. In *ACM Siggraph Computer Graphics*, vol. 26, pp. 177–184. ACM.

H. Huang, et al. (2017). 'Learning local shape descriptors from part cor-

respondences with multiview convolutional networks'. *ACM Transactions on Graphics (TOG)* **37**(1):1–14.

J. Huang, et al. (2006). 'Subspace gradient domain mesh deformation'. In *ACM Transactions on Graphics (TOG)*, vol. 25, pp. 1126–1134. ACM.

Q.-X. Huang, et al. (2008). 'Non-rigid registration under isometric deformations'. In *Computer Graphics Forum*, vol. 27, pp. 1449–1457. Wiley Online Library.

S. Irani & P. Raghavan (1999). 'Combinatorial and experimental results for randomized point matching algorithms'. *Computational Geometry* **12**(1-2):17–31.

G. Irving, et al. (2004). 'Invertible finite elements for robust simulation of large deformation'. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 131–140. Eurographics Association.

A. Jacobson (2013). *Algorithms and interfaces for real-time deformation of 2d and 3d shapes*. Ph.D. thesis.

A. Jacobson, et al. (2011). 'Bounded biharmonic weights for real-time deformation.'. *ACM Trans. Graph.* **30**(4):78.

A. Jacobson, et al. (2010). 'Mixed finite elements for variational surface modeling'. In *Computer Graphics Forum*, vol. 29, pp. 1565–1574. Wiley Online Library.

A. Jacobson, et al. (2012). 'Smooth Shape-Aware Functions with Controlled Extrema'. In *Computer Graphics Forum*, vol. 31, pp. 1577–1586. Wiley Online Library.

V. Jain, et al. (2007). 'Non-rigid spectral correspondence of triangle meshes'. *International Journal of Shape Modeling* **13**(01):101–124.

T. Jiang, et al. (2017). 'Consistent as-similar-as-possible non-isometric surface registration'. *The Visual Computer* **33**(6-8):891–901.

Y. Jin, et al. (2014). 'Remeshing-assisted Optimization for Locally In-

jective Mappings'. In *Computer Graphics Forum*, vol. 33, pp. 269–279. Wiley Online Library.

A. E. Johnson (1997). 'Spin-images: a representation for 3-D surface matching' .

A. E. Johnson & M. Hebert (1999). 'Using spin images for efficient object recognition in cluttered 3D scenes'. *IEEE Transactions on pattern analysis and machine intelligence* **21**(5):433–449.

P. Joshi, et al. (2007). 'Harmonic coordinates for character articulation'. In *ACM Transactions on Graphics (TOG)*, vol. 26, p. 71. ACM.

T. Ju, et al. (2005). 'Mean value coordinates for closed triangular meshes'. In *ACM Transactions on Graphics (TOG)*, vol. 24, pp. 561–566. ACM.

O. Kin-Chung Au, et al. (2010). 'Electors voting for fast automatic shape correspondence'. In *Computer Graphics Forum*, vol. 29, pp. 645–654. Wiley Online Library.

L. Kobbelt, et al. (1998). 'Interactive multi-resolution modeling on arbitrary meshes'. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 105–114. ACM.

L. Kobbelt, et al. (1999). 'Multiresolution hierarchies on unstructured triangle meshes'. *Computational Geometry* **14**(1-3):5–24.

S. Z. Kovalsky, et al. (2015). 'Large-scale bounded distortion mappings.'. *ACM Trans. Graph.* **34**(6):191–1.

A. Kovnatsky, et al. (2015). 'Functional correspondence by matrix completion'. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 905–914.

A. Kovnatsky, et al. (2013). 'Coupled quasi-harmonic bases'. In *Computer Graphics Forum*, vol. 32, pp. 439–448. Wiley Online Library.

V. Kraevoy & A. Sheffer (2004). 'Cross-parameterization and compatible remeshing of 3D models'. In *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 861–869. ACM.

V. Kraevoy & A. Sheffer (2005). 'Template-Based Mesh Completion.'. In *Symposium on Geometry Processing*, vol. 385, pp. 13–22.

A. Kurenkov, et al. (2018). 'Deformnet: Free-form deformation network for 3d shape reconstruction from a single image'. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 858–866. IEEE.

H. Laga, et al. (2006). 'Spherical wavelet descriptors for content-based 3D model retrieval'. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pp. 15–15. IEEE.

A. Lee, et al. (2000). 'Displaced subdivision surfaces'. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 85–94. ACM Press/Addison-Wesley Publishing Co.

Z. Levi & C. Gotsman (2015). 'Smooth rotation enhanced as-rigid-as-possible mesh animation'. *IEEE Transactions on Visualization and Computer Graphics* **21**(2):264–277.

H. Li, et al. (2008). 'Global Correspondence Optimization for Non-Rigid Registration of Depth Scans'. In *Computer graphics forum*, vol. 27, pp. 1421–1430. Wiley Online Library.

K. Li, et al. (2018). 'Robust non-rigid registration with reweighted position and transformation sparsity'. *IEEE transactions on visualization and computer graphics* **25**(6):2255–2269.

X.-Y. Li & S.-M. Hu (2013). 'Poisson coordinates'. *IEEE Transactions on visualization and computer graphics* **19**(2):344–352.

M. Liao, et al. (2009). 'Modeling deformable objects from a single depth camera'. In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 167–174. IEEE.

Y. Lipman (2012). 'Bounded distortion mapping spaces for triangular meshes'. *ACM Transactions on Graphics (TOG)* **31**(4):108.

Y. Lipman, et al. (2007). 'GPU-assisted positive mean value coordinates for mesh deformations'. In *Symposium on geometry processing*.

Y. Lipman, et al. (2008). 'Green coordinates'. In *ACM Transactions on Graphics (TOG)*, vol. 27, p. 78. ACM.

Y. Lipman, et al. (2004). 'Differential coordinates for interactive mesh editing'. In *Shape Modeling Applications, 2004. Proceedings*, pp. 181–190. IEEE.

Y. Lipman, et al. (2005). 'Linear rotation-invariant coordinates for meshes'. *ACM Transactions on Graphics (TOG)* **24**(3):479–487.

R. Liu, et al. (2009). 'A part-aware surface metric for shape analysis'. In *Computer Graphics Forum*, vol. 28, pp. 397–406. Wiley Online Library.

T. Liu, et al. (2016). 'Fast and Robust Inversion-Free Shape Manipulation'. In *Computer Graphics Forum*, vol. 35, pp. 1–11. Wiley Online Library.

R. MacCracken & K. I. Joy (1996). 'Free-form deformations with lattices of arbitrary topology'. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 181–188. ACM.

S. Manay, et al. (2006). 'Integral invariants for shape matching'. *IEEE Transactions on pattern analysis and machine intelligence* **28**(10):1602–1618.

H. Maron, et al. (2016). 'Point registration via efficient convex relaxation'. *ACM Transactions on Graphics (TOG)* **35**(4):73.

M. Meyer, et al. (2003). 'Discrete differential-geometry operators for triangulated 2-manifolds'. In *Visualization and Mathematics III*, pp. 35–57. Springer.

C. Moenning & N. A. Dodgson (2003). 'Fast marching farthest point sampling'. Tech. rep., University of Cambridge, Computer Laboratory.

M. Müller, et al. (2005). 'Meshless deformations based on shape matching'. In *ACM transactions on graphics (TOG)*, vol. 24, pp. 471–478. ACM.

M. Ovsjanikov, et al. (2012). 'Functional maps: a flexible representation

of maps between shapes'. *ACM Transactions on Graphics (TOG)* **31**(4):1–11.

D. Panozzo, et al. (2010). 'Efficient multiscale curvature and crease estimation'. *Proceedings of Computer Graphics, Computer Vision and Mathematics (Brno, Czech Rapubic* **1**(6).

C. Papazov & D. Burschka (2011). 'Deformable 3D shape registration based on local similarity transforms'. In *Computer Graphics Forum*, vol. 30, pp. 1493–1502. Wiley Online Library.

M. Pauly, et al. (2003). 'Multi-scale feature extraction on point-sampled surfaces'. In *Computer graphics forum*, vol. 22, pp. 281–289. Wiley Online Library.

M. Pauly, et al. (2005). 'Example-based 3D scan completion'. In *Symposium on Geometry Processing*, no. EPFL-CONF-149337, pp. 23–32.

G. Peyré & L. D. Cohen (2006). 'Geodesic remeshing using front propagation'. *International Journal of Computer Vision* **69**(1):145.

U. Pinkall & K. Polthier (1993). 'Computing discrete minimal surfaces and their conjugates'. *Experimental mathematics* **2**(1):15–36.

J. Pokrass, et al. (2013). 'Sparse modeling of intrinsic correspondences'. In *Computer Graphics Forum*, vol. 32, pp. 459–468. Wiley Online Library.

R. Poranne & Y. Lipman (2014). 'Provably good planar mappings'. *ACM Transactions on Graphics (TOG)* **33**(4):76.

E. Richardson, et al. (2017). 'Learning detailed face reconstruction from a single image'. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1259–1268.

A. R. Rivers & D. L. James (2007). 'FastLSM: fast lattice shape matching for robust real-time deformation'. *ACM Transactions on Graphics (TOG)* **26**(3):82.

L. I. Rudin, et al. (1992). 'Nonlinear total variation based noise removal algorithms'. *Physica D: nonlinear phenomena* **60**(1-4):259–268.

S. Rusinkiewicz & M. Levoy (2001). 'Efficient variants of the ICP algorithm'. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pp. 145–152. IEEE.

C. Schüller, et al. (2013). 'Locally injective mappings'. In *Computer Graphics Forum*, vol. 32, pp. 125–135. Wiley Online Library.

T. W. Sederberg & S. R. Parry (1986). 'Free-form deformation of solid geometric models'. *ACM SIGGRAPH computer graphics* **20**(4):151–160.

R. Setaluri, et al. (2014). 'Fast grid-based nonlinear elasticity for 2D deformations'. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 67–76. Eurographics Association.

X. Shi, et al. (2007). 'Mesh puppetry: cascading optimization of mesh deformation with inverse kinematics'. In *ACM Transactions on Graphics (TOG)*, vol. 26, p. 81. ACM.

H. Si (2015). 'TetGen, a Delaunay-based quality tetrahedral mesh generator'. *ACM Transactions on Mathematical Software (TOMS)* **41**(2):11.

K. Singh & E. Fiume (1998). 'Wires: a geometric deformation technique'. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 405–414. ACM.

K. Singh & E. Kokkevis (2000). 'Skinning Characters using Surface Oriented Free-Form Deformations.'. In *Graphics interface*, vol. 2000, pp. 35–42.

O. Sorkine & M. Alexa (2007). 'As-rigid-as-possible surface modeling'. In *Symposium on Geometry processing*, vol. 4, p. 30.

O. Sorkine, et al. (2004). 'Laplacian surface editing'. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pp. 175–184. ACM.

A. Stomakhin, et al. (2012). 'Energetically consistent invertible elastic-

ity'. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 25–32. Eurographics Association.

R. W. Sumner, et al. (2007). 'Embedded deformation for shape manipulation'. In *ACM Transactions on Graphics (TOG)*, vol. 26, p. 80. ACM.

J. Sun, et al. (2009). 'A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion'. In *Computer graphics forum*, vol. 28, pp. 1383–1392. Wiley Online Library.

G. K. Tam, et al. (2013). 'Registration of 3D point clouds and meshes: a survey from rigid to nonrigid'. *IEEE transactions on visualization and computer graphics* **19**(7):1199–1217.

D. Terzopoulos, et al. (1987). 'Elastically deformable models'. In *ACM Siggraph Computer Graphics*, vol. 21, pp. 205–214. ACM.

A. Tevs, et al. (2011). 'Intrinsic shape matching by planned landmark sampling'. In *Computer Graphics Forum*, vol. 30, pp. 543–552. Wiley Online Library.

A. Tevs, et al. (2009). 'Isometric registration of ambiguous and partial data'. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1185–1192. Ieee.

G. Turk & M. Levoy (1994). 'Zippered polygon meshes from range images'. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pp. 311–318. ACM.

W. von Funck, et al. (2006). 'Vector field based shape deformations'. *ACM Transactions on Graphics (TOG)* **25**(3):1118–1125.

H. Wang, et al. (2018). 'Learning 3d keypoint descriptors for non-rigid shape matching'. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19.

W. Wang, et al. (2019). '3dn: 3d deformation network'. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1038–1046.

Y. Wang, et al. (2020). 'MGCN: Descriptor learning using multiscale GCNs'. *arXiv preprint arXiv:2001.10472* .

O. Weber, et al. (2011). 'A complex view of barycentric mappings'. In *Computer Graphics Forum*, vol. 30, pp. 1533–1542. Wiley Online Library.

O. Weber, et al. (2007). 'Context-Aware Skeletal Shape Deformation'. In *Computer Graphics Forum*, vol. 26, pp. 265–274. Wiley Online Library.

T. Weise, et al. (2009). 'Face/off: Live facial puppetry'. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation*, pp. 7–16. ACM.

M. Werlberger, et al. (2009). 'Anisotropic Huber-L1 Optical Flow.'. In *BMVC*, vol. 1, p. 3.

W. Xu, et al. (2015). 'Deformable 3D fusion: From partial dynamic 3D observations to complete 4D models'. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2183–2191.

S. Yamazaki, et al. (2013). 'Non-rigid shape registration using similarity-invariant differential coordinates'. In *3D Vision-3DV 2013, 2013 International Conference on*, pp. 191–198. IEEE.

J. Yang, et al. (2015). 'Sparse Non-rigid Registration of 3D Shapes'. In *Computer Graphics Forum*, vol. 34, pp. 89–99. Wiley Online Library.

I.-C. Yeh, et al. (2011). 'Template-based 3d model fitting using dual-domain relaxation'. *IEEE Transactions on Visualization and Computer Graphics* **17**(8):1178–1190.

Y. Yoshiyasu, et al. (2014). 'As-Conformal-As-Possible Surface Registration'. In *Computer Graphics Forum*, vol. 33, pp. 257–267. Wiley Online Library.

T. Yu, et al. (2018). 'Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor'. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7287–7296.

Y. Yu, et al. (2004). 'Mesh editing with poisson-based gradient field manipulation'. In *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 644–651. ACM.

C. Zach, et al. (2007). 'A duality based approach for realtime TV-L 1 optical flow'. In *Joint Pattern Recognition Symposium*, pp. 214–223. Springer.

R. Zayer, et al. (2005). 'Harmonic guidance for surface deformation'. In *Computer Graphics Forum*, vol. 24, pp. 601–609. Wiley Online Library.

H. Zhang, et al. (2008). 'Deformation-Driven Shape Correspondence'. In *Computer Graphics Forum*, vol. 27, pp. 1431–1439. Wiley Online Library.